# Distributed Source Coding of Multi-View Images

by
Nicolas Gehrig
*Dipl.-Ing. (M.Sc.)*

A Thesis submitted in fulfilment of requirements for the degree of
Doctor of Philosophy of University of London and
Diploma of Imperial College

Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London
University of London
2007

# List of Publications

The following publications were produced during the time of this work:

1. Nicolas Gehrig and Pier Luigi Dragotti. *Distributed Compression of the Plenoptic Function.* IEEE International Conference on Image Processing (ICIP 2004), October 2004, Singapore.

2. Nicolas Gehrig and Pier Luigi Dragotti. *Distributed Compression in Camera Sensor Networks.* IEEE Workshop on Multimedia Signal Processing (MMSP 2004), September 2004, Siena, Italy.

3. Nicolas Gehrig and Pier Luigi Dragotti. *Symmetric and a-symmetric Slepian-Wolf codes with systematic and non-systematic linear codes.* IEEE Communications Letters, vol. 9 (1), pp. 61-63, January 2005.

4. Nicolas Gehrig and Pier Luigi Dragotti. *On Distributed Compression in Dense Camera Sensor Networks.* Picture Coding Symposium (PCS 2004), December 2004, San Francisco, USA.

5. Nicolas Gehrig and Pier Luigi Dragotti. *DIFFERENT - DIstributed and Fully Flexible image EncodeRs for camEra sensor NeTwork.* IEEE International Conference on Image Processing (ICIP 2005), September 2005, Genova, Italy.

6. Nicolas Gehrig and Pier Luigi Dragotti. *Distributed Sampling and Compression of Scenes with Finite Rate of Innovation in Camera Sensor Networks.*

Data Compression Conference (DCC 2006), March 2006, Snowbird, Utah.

7. Nicolas Gehrig and Pier Luigi Dragotti. *Distributed compression of multi-view images using a geometrical coding approach.* IEEE International Conference on Image Processing (ICIP 2007), September 2007, San Antonio, Texas.

8. Nicolas Gehrig and Pier Luigi Dragotti. *Geometry-driven distributed compression of the Plenoptic Function: Performance bounds and constructive algorithms.* Submitted to IEEE Transactions on Image Processing, January 2008.

# Abstract

In this thesis, we consider camera sensor networks where numbers of sensors are deployed in order to visually acquire a given scene from different viewing positions. Each sensor consists of a self-powered wireless device containing a digital camera, a processing unit with memory and some communication capabilities. The main task for the sensors is to acquire and process their observations of the scene, and then transmit some information to a common receiver in order to guarantee the best possible reconstruction of the different views by the decoder. Due to the spatial proximity of the different cameras, acquired images can be highly dependent. Since the sensors have limited power and communication resources, the inter-sensor dependencies have to be exploited in order to reduce the transmission of redundant information.

Our work addresses the problem of distributed compression of multi-view images. The main challenge in such a compression scheme is that the different encoders have to exploit the correlation in the visual information without being allowed to communicate between themselves. This problem, known as *Distributed Source Coding*, requires that the encoders have some knowledge about the correlation structure of the visual information acquired from the different cameras. This correlation, which is related to the structure of the plenoptic function, can be estimated using geometrical information such as the position of the cameras and some bounds on the location of the objects.

We first propose a lossless distributed compression scheme for simple scenes that takes advantage of this geometrical information in order to reduce the overall transmission rate from the sensors to a common central receiver. Our approach allows for a flexible allocation of the bit-rates amongst the encoders (i.e., it can cover the entire Slepian-Wolf achievable rate region) and can be made resilient to a fixed number of occlusions. Then, the fundamental trade-offs between the reconstruction fidelity, the number and location of the cameras and the overall compression rate are analysed. A practical algorithm based on quadtree decomposition for distributed lossy compression of real multi-view images is then proposed. Finally, we also show that our approach can be extended to provide a general solution to the problem of distributed compression of correlated binary sources.

# Acknowledgments

***Hitotsu! Doryoku No Seishin O Yashinau Koto!***
*(Cultivate a spirit of endeavour and perseverance - Dojo Kun, line 3)*
*- Sensei Gichin FUNAKOSHI, Founder of Shotokan Karate, 1868-1957.*

I would like to express my deepest gratitude to my supervisor, Dr. Pier Luigi Dragotti, for his time, conductive advice, and willingness to share his insight and wisdom. His constant support and guidance during all the time of my PhD has lead me to the right way. I would also like to thank him for his patience during all the months that I needed to finish my writing-up from Switzerland because I had already started a new job...

There are many other people that have played an important role for me during these three to four years of research. I will try to thank some of them personally in the next few lines and apologize for all the others that would have deserved to be mentioned here.

Imperial College has allowed me to meet great people. I have to thank Fotis who was the first to introduce me to this completely new environment, and also took the time to show me that our campus was not the only interesting place in London. Over the years, I met other great friends as well and I would like to thank

Nick (by the way, thanks man for printing my thesis), Loic, Jesse, Pancham, Jon, Eagle, Vinesh and all the others for the great time I have had with them both in the lab and outside of it.

I would like to thank Prof. Kannan Ramchandran who gave me the opportunity to work in his lab at UC Berkeley for three months. My interactions with all the people I met in California have been extremely enriching and have allowed me to experience completely different approaches to research and campus life.

I have been very lucky to find the Imperial College Shotokan Karate Club, and I would like to thank Sensei Jim Lewis for his warm welcome and his constant motivation to conduct his karate classes. His expertise and the quality of his teaching was much higher than what I could have dreamed of.

I want to thank all my friends from Switzerland (Guillaume, Stefano, Ron, Cédric,...) that have come to visit me in London and have made my travels back home a real enjoyment every time.

Obviously, I am infinitely thankful to all my family and in particular to my parents. They have always supported me in my decisions and have done all they could do to help me reach my objectives.

Finally, you would not be reading this thesis if it was not for one person who convinced me to jump on the opportunity and move to London, whilst being aware that it would make her life much more complicated. I will probably never find the words to express all my gratitude to her. At least, I can start with a: *Merci beaucoup Mélanie.*

# Contents

# List of Figures

# List of Tables

# Statement of Originality

As far as the author is aware, the following aspects of the thesis are believed to be original contributions:

- We propose a distributed source coding approach for simple multi-view images that uses some information about the epipolar geometry to efficiently encode the position of objects on the different views. We show how the correlation between the different images can be estimated and then exploited in a distributed manner. The proposed approach can be made resilient to visual occlusions and allows for a flexible allocation of the transmission rates amongst the encoders.

- We study some fundamental trade-offs in camera sensor networks for ideal scenarios. In particular, we show that if the observed scene is of finite rate of innovation (FRI) and can be parameterized with a set of polygons of polynomial intensities, then we can highlight an exact bit-conservation principle for the rate-distortion behaviour at the decoder. In other words, we show that the reconstruction quality does not depend on the number of sensors used for the transmission but only on the total bit-rate transmitted to the receiver.

- We present a practical distributed compression approach for real multi-view images. We show that an existing tree-based compression scheme can be used at each encoder and propose a coding approach to efficiently exploit the remaining multi-view correlation before transmitting the data to the receiver.

We use a piecewise polynomial model to represent the images and show that our approach still outperforms an independent encoding when the images do not fully satisfy the correlation model.

- We show that our distributed coding strategy using epipolar geometry can be intuitively extended to provide a flexible distributed source coding scheme for correlated binary sources. In particular, we demonstrate that this approach can be used to achieve any point on the Slepian-Wolf achievable rate region.

# Abbreviations

**CEO:** Chief Executive Officer

**CSN:** Camera Sensor Network

**DSC:** Distributed Source Coding

**EPI:** Epipolar Plane Image

**H.26x:** A video codec family

**IBR:** Image-Based Rendering

**LDPC:** Low-Density Parity-Check

**LFR:** Light-Field Rendering

**MPEG:** Moving Picture Experts Group

**MSE:** Mean Squared Error

**PF:** Plenoptic Function

**PSNR:** Peak Signal-to-Noise Ratio

**R-D:** Rate-Distortion

**RGB:** Red-Green-Blue

**SNR:** Signal-to-Noise Ratio

**S-W:** Slepian-Wolf

**VRML:** Virtual Reality Markup Language

**W-Z:** Wyner-Ziv

# Symbols

| | | |
|---|---|---|
| $X, Y$ | : | Two random sources (binary sources, object's positions, ...) |
| $R_x, R_y$ | : | Encoding rates for sources $X$ and $Y$ |
| $n$ | : | An integer |
| $x^n, y^n$ | : | Sequences of $n$ successive outcomes for $X$ and $Y$ |
| $\mathcal{X}^n, \mathcal{Y}^n$ | : | Spaces of all length-$n$ sequences for $X$ and $Y$ |
| $x_i, y_i$ | : | $i^{th}$ outcomes of $X$ and $Y$ |
| $P_X(x_i), P_Y(y_i)$ | : | Probability distributions of $X$ and $Y$ |
| $P_{X,Y}(x_i, y_i)$ | : | Joint probability distribution of $X$ and $Y$ |
| $A_\epsilon^{(n)}$ | : | Set of all jointly typical pairs $(x^n, y^n)$ |
| $i, j, k$ | : | Indices |
| $H(X)$ | : | Entropy of $X$ |
| $H(X, Y)$ | : | Joint entropy of $X$ and $Y$ |
| $H(X\|Y)$ | : | Conditional entropy of $X$ given $Y$ |
| $I(X, Y)$ | : | Mutual information of $X$ and $Y$ |
| $D$ | : | Distortion value |
| $\hat{X}$ | : | Estimate of $X$ |
| $E\{\cdot\}$ | : | Expectation operator |
| $d(\cdot, \cdot)$ | : | Distortion metric |
| $R(D)$ | : | Rate-distortion function |

$R^*_{WZ}(D)$ : Optimal rate-distortion function for the W-Z problem

$(\theta, \phi)$ : Pan-tilt orientation of the camera

$\omega$ : lightwave frequency

$\tau$ : time

$(V_x, V_y, V_z)$ : 3-D position of the camera in space

$I_{PF}$ : Intensity of the light

$(s, t)$ : Coordinates of the focal plane

$(u, v)$ : Coordinates of the retinal plane

$\alpha$ : Distance between two cameras

$N$ : Number of sources (cameras, sensors, ...)

$z$ : Depth of an object

$(z_{min}, z_{max})$ : Minimum and maximum depths

$f$ : focal length

$\Delta$ : Disparity

$(\Delta_{min}, \Delta_{max})$ : Minimum and maximum disparities

$\delta$ : Length of disparity range

$R_{min}$ : Number of least significant bits

$S(\cdot)$ : A subset

$\bar{S}(\cdot)$ : The complementary subset

$r$ : constant rate

$\Theta(\cdot)$ : Order operator

$\{f_1(t), f_2(t)\}$ : Pair of correlated 1-D signals

$T$ : Support of a signal

$A$ : Maximum amplitude

$L$ : Number of pieces

$Q$ : Maximum degree of polynomial

$t_{n,i}$ : $i^{th}$ discontinuity of the $n^{th}$ view

$N_{min}$ : Minimum number of views

$O_{max}$ : Maximum number of occlusions

$R_t$ : Rate to encode discontinuities

$R_p$ : Rate to encode polynomials

$R_{t_{SW}}$ : Rate to encode at minimum S-W rate

$\gamma_s, G$ : Constants

$R_{tot}$ : Total rate to encode all the sources

$\{c_0, c_1, c_2, c_3\}$ : Constants

$\{D_A, D_B, D_C\}$ : Distortion values

$J_{max}$ : Maximum tree depth

$\lambda$ : Operating slope

$R_{Tree}$ : Rate to encode pruned tree

$R_{LeafJointCoding}$ : Rate to encode the joining info

$R_{Leaves}$ : Rate to encode the set of polynomials

$J_\Delta$ : Constant tree depth

$R_\Delta$ : Number of nodes with depth smaller than $J_\Delta$

$\{V_1(x,y), V_2(x,y)\}$ : Pair of correlated 2-D signals

$\{\mathcal{L}_i(x,y)\}$ : 2-D orthonormal Legendre basis

$l_i$ : Set of Legendre coefficients

$\hat{l}_i$ : Set of quantized Legendre coefficients

$\mathcal{C}$ : Linear code of length $n$ and rank $k$

$\mathbf{H}$ : Parity check matrix for $\mathcal{C}$

$\mathbf{G}$ : Generator matrix for $\mathcal{C}$

$C$ : Channel capacity

$\oplus$ : Binary addition operator

$d_H(\cdot,\cdot)$ : Hamming distance operator

$s_x$ : Syndrome of $x$

$\mathbf{I}_k$ : Identity matrix of size $k \times k$

$(\cdot)^T$ : Transpose operator

$e_d^k$ : The $k$ first bits of difference pattern $e_d$

# Chapter 1

# Introduction

## 1.1 Motivations

RECENT advances in sensor network technology [4] are radically changing the way in which we sense, process and transport signals of interest. The usual one-to-one scenario where information acquired from a unique source is encoded and then transmitted through a communication channel to a unique receiver is today well understood, but cannot provide satisfactory answers for the many-to-many scenario that sensor networks bring on the table. Phenomena of interests that are acquired by sensor networks are distributed in space and may exhibit very peculiar structures. The acquisition of a given phenomenon is done by fusing all the local measurements of each sensor. These observations can be highly correlated and depend on the structure of the phenomenon and the actual spatial deployment of the sensors.

A wireless sensor network consists of numbers of small self-powered devices that have embedded sensing, processing and communication capabilities. The size, the limitation of power resources and the necessity to maintain cheap prices are usually the main constraints with such systems. This obliges the developers to consider several trade-offs when designing sensor network systems. These trade-offs

typically involves acquisition accuracy, computational power, capacity of memory, transmission power, delay and battery life duration.

In our work, we focus our attention on camera sensor networks (CSN) where each sensor is equipped with a digital camera and acquires images of a scene of interest from a certain viewing position. The phenomenon observed by the different sensors corresponds therefore to the visual information emanating from the scene and can be represented with the Plenoptic Function [3]. The study of the structure of this function has already led to the development of sampling strategies [9], and can also be used to understand the geometrical relationships between the different views. Thanks to the recent technological advances in cameras and networked sensor platforms, the variety of available CSN equipments is becoming wider and wider and constantly improves the opportunities for new applications and set-ups. In [32], the latest technology trends in cameras and sensor platforms are presented.

In any sensor network, as the density of sensors increases, the different observations of the phenomenon may become more correlated. If we assume that each sensor has to send independently its measurements to a common central receiver, the amount of redundant information transmitted may become very important. This waste of communication and power resources could typically be unacceptable in most sensor network applications. The correlation, which is directly related to the physical properties of the phenomenon, should therefore be exploited at each sensor to compress the data before transmitting it to the receiver. A first solution could be to let the sensors perform a joint compression of their observations in a collaborative way. However, this solution would require a complex inter-sensor communication system that would consume most of the sensors resources and could even simply not be available in many cases. We therefore need to find a way to exploit the correlation at the encoders without allowing them to communicate between themselves.

In 1973, Slepian and Wolf [72] showed that independent encoding of correlated

sources can be, in theory, as efficient as joint encoding, as long as the encoders know the correlation structure between the sources. In other words, if the sensors have some information about the properties of the phenomenon they observe, they can estimate the correlation and use it to perform a local compression that could reach the same global rate-distortion performance as a joint encoding. This surprising result is at the root of the theoretical foundation of Distributed Source Coding (DSC) that has recently led to practical coding approaches for correlated sources [52]. However, these practical coding approaches are based on channel coding principles and are only suitable for certain correlation models.

In this thesis, our aim is to show that the correlation in the visual data acquired by a camera sensor network can be estimated using some limited geometrical information about the scene and the position of the cameras. This correlation is related to the structure of the plenoptic function and cannot be correctly exploited with existing DSC approaches based on channel codes. We therefore propose a specific distributed coding strategy for simple synthetic scenes that can take advantage of this estimated correlation to reduce the amount of data to be transmitted from the sensors to the receiver. We show that our approach allows for a flexible allocation of the transmission bit-rates amongst the encoders and can be made resilient to the problem of visual occlusions. Then, we study the fundamental trade-offs between the number of sensors and the reconstruction fidelity at the decoder and highlight a bit-conservation principle. We propose then a practical distributed compression algorithm based on quadtree decomposition for real multi-view images. Finally, we show that our coding approach can be intuitively extended to perform distributed compression of correlated binary sources.

## 1.2 Related Work

The representation and compression of 3D scenes have been widely studied in the last decade. The usual model-based approaches used in standards such as VRML [76] are slowly being replaced by new image-based representation techniques [71], where the 3D information of a scene is simply represented by a collection of 2D images obtained from different viewing positions. The relationship between these different views is governed by the laws of projective geometry [16, 27] and can be used to develop efficient rendering techniques [25, 34]. On the compression side, significant efforts have been put on the development of stereo image coders [6, 8, 51], that mainly rely on block-based disparity compensation techniques. These approaches have recently led to the development of more advanced multi-view image and video encoders [7, 42, 48]. All these compression techniques rely on a joint encoding of the different views to exploit the correlation in the data acquired by the different cameras. In order to exploit this correlation without allowing the cameras to collaborate, specific distributed source coding techniques should be developed.

Inspired by the theoretical results obtained by Slepian and Wolf for the lossless case [72] and by Wyner and Ziv for the lossy case [85], Pradhan and Ramchandran proposed a first constructive coding approach based on channel coding principles for distributed source coding of correlated sources [52]. Practical designs based on advanced channel codes such as Turbo [1, 19, 20, 33, 35, 40] and LDPC [11, 37, 38, 62, 63, 74, 90] codes have since been proposed in the last few years. All these coding approaches can closely approach the theoretical bounds for different binary correlation models, but cannot be efficiently used to exploit the correlation structure of the visual information obtained from a multi-camera system. Nevertheless, several researchers have recently used these approaches with correlated visual information to develop distributed video coding algorithms [24, 57, 66, 89] (see Section 2.3.2 for

details), and distributed multi-view image coding schemes [2, 30, 36, 77–79, 94].

Our main contribution in this thesis is to propose a distributed compression approach for multi-view images that can truly exploit the correlation in the visual information which is related to the structure of the plenoptic function [3], without requiring the use of complex channel coding techniques. An outline of the thesis is proposed in the next section.

## 1.3 Thesis Outline

The thesis is organized as follows:

- In Chapter 2, we present a review of distributed source coding (DSC). We first introduce the theoretical foundations derived from information theory principles in the early 70's. Then, we present the design of practical coders based on existing channel coding techniques. Finally we highlight some recent applications of distributed source coding, such as distributed compression in wireless sensor networks, distributed video and multi-view coding, and joint source-channel coding.

- In Chapter 3, we propose a distributed compression strategy for simple synthetic scenes that can truly exploit the geometrical correlation available in multi-view images. We first introduce the plenoptic function which can be used to represent the geometrical dependencies between the different views. Then, we propose a coding scheme that can be used to efficiently encode the positions of the objects on the different views in a fully distributed manner. Finally, we show that our approach can be robust to the problem of visual occlusions and present simulation results. Notice that the preliminary results we present in this chapter will be used in the following chapters to study some

fundamental trade-offs in camera sensor networks (Chapter 4), develop a distributed compression algorithm for real multi-view images (Chapter 5), and propose a distributed source coding approach for correlated binary sources (Chapter 6).

- In Chapter 4, we study some fundamental trade-offs in camera sensor networks. In particular, we show that if the observed scene is of finite rate of innovation (FRI) and can be represented exactly with polygons of polynomial intensities, an exact sampling strategy can be proposed. Moreover, we derive rate-distortion bounds for the reconstructions of the views at the decoder and show that an exact bit-conservation principle exists. In other words, the quality of the reconstructed views only depends on the total transmission bit-rate and not on the number of sensors involved.

- In Chapter 5, we propose a practical algorithm for distributed compression of real multi-view images. We use a piecewise polynomial model to represent the different views and show that a tree-based coding approach recently proposed in the literature can be extended to take advantage of our distributed scheme presented in Chapter 3. We first give a detailed description of our distributed algorithm for the 1-D case using binary tree segmentation, and then show how it can be extended to the 2-D case using a quadtree approach. Finally, we present some simulation results obtained on real multi-view images.

- In Chapter 6, we propose a distributed source coding approach for correlated binary sources directly inspired by the scheme presented in Chapter 3. In particular, we show that this intuitive approach can cover the entire Slepian-Wolf achievable rate region, and present a simple example to give the correct intuition.

- In Chapter 7, we give a summary of the thesis and highlight a selection of

interesting research directions that should be investigated in future works.

# Chapter 2

# Distributed Source Coding

## 2.1 Theoretical Background

Consider a communication system where two discrete memoryless random sources $X$ and $Y$ must be encoded and transmitted to a common receiver. Assume that $X$ and $Y$ are encoded at rates $R_x$ and $R_y$ respectively. We know from Shannon's information theory principles that these rates are sufficient to perform noiseless coding if they are equal or greater than the entropy of the sources (i.e. $R_x \geq H(X)$ and $R_y \geq H(Y)$), where the entropy of a discrete memoryless source $X$ with $n$ possible outcomes $(x_1, \ldots, x_n)$ is defined as:

$$H(X) = -\sum_{i=1}^{n} p_X(x_i) \log_2 p_X(x_i) \,, \tag{2.1}$$

where $p_X(x_i)$ is the probability of the $i^{\text{th}}$ possible outcome for $X$, and thus satisfies the following equality: $\sum_{i=1}^{n} p_X(x_i) = 1$.

Assume now that the sources $X$ and $Y$ are correlated and have a joint probability distribution given by $p_{X,Y}(x, y)$. We know that if they can be jointly encoded, a total rate corresponding to the joint entropy of the sources is sufficient to perform

noiseless coding (i.e. $R_x + R_y \geq H(X,Y)$), where the joint entropy of $X$ and $Y$ is defined as:

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{X,Y}(x_i,y_j)\log_2 p_{X,Y}(x_i,y_j)\,. \tag{2.2}$$

Now assume that these two sources are physically separated and cannot communicate with each other. A joint encoding of the sources is therefore not possible in this case. Nevertheless, Slepian and Wolf [72] showed in 1973 that noiseless encoding of $X$ and $Y$ is still achievable in this context if $R_x \geq H(X|Y)$, $R_y \geq H(Y|X)$ and $R_x + R_y \geq H(X,Y)$, where $H(X|Y)$ corresponds to the conditional entropy of $X$ given $Y$ and is defined as:

$$H(X|Y) = H(X,Y) - H(Y)\,. \tag{2.3}$$

This surprising result gives the theoretical foundation of *Distributed Source Coding* (DSC). It shows that there is in theory no loss in terms of overall rate even though the encoders are separated (see Figure 2.1).

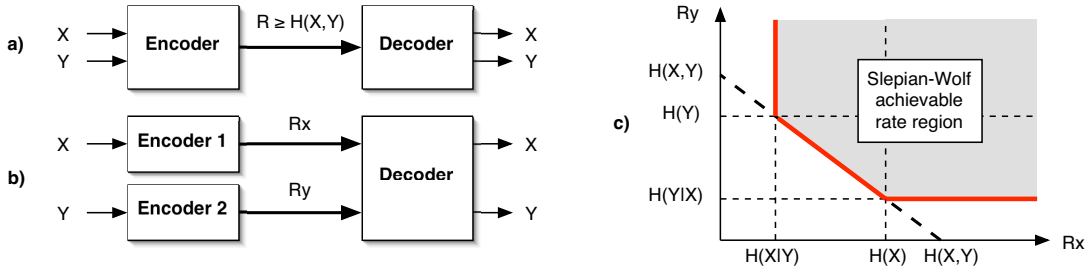


**Figure 2.1: (a) Joint source coding. (b) Distributed source coding. The Slepian-Wolf theorem (1973) states that a combined rate of $H(X,Y)$ remains sufficient even if the correlated sources are encoded separately. (c) The achievable rate region is given by: $R_x \geq H(X|Y)$, $R_y \geq H(Y|X)$ and $R_x + R_y \geq H(X,Y)$.**

The proof of the achievability of the Slepian-Wolf theorem is asymptotical, non-constructive, and is based on random binning [13]. *Binning*, which refers to the partitioning of the space of all possible outcomes of a random source into different

subsets, is a key concept in DSC. The basic idea of the proof is to partition the space of all *length-n* sequences by randomly assigning every $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$ to one of $2^{nR_x}$ and $2^{nR_y}$ bins respectively. These assignments ($f_x : \mathcal{X}^n \rightarrow \{1, \ldots, 2^{nR_x}\}$ and $f_y : \mathcal{Y}^n \rightarrow \{1, \ldots, 2^{nR_y}\}$) are known by both the encoders and the decoder. The encoding process at each source consists then in sending only the index of the bin to which the current sequence belongs. At the decoder, given the pair of indices $(i_x, i_y)$, the original pair of sequences can be retrieved if there is one and only one pair $(x^n, y^n)$ such that $f_x(x^n) = i_x$, $f_y(y^n) = i_y$, and $(x^n, y^n) \in A_\epsilon^{(n)}$, where $A_\epsilon^{(n)}$ is the set of all jointly typical pairs of sequences [13]. Otherwise, an error is declared. The rest of the proof shows that the probability of decoding error can be made arbitrarily small by increasing the sequence length $n$ when $R_x \geq H(X|Y)$, $R_y \geq H(Y|X)$ and $R_x + R_y \geq H(X, Y)$.

An extension of the Slepian-Wolf result to the lossy case (with continuous sources) was proposed by Wyner and Ziv in [85]. They addressed a particular case of Slepian-Wolf coding corresponding to the rate point $(R_x, R_y) = (H(X|Y), H(Y))$, also known as *source coding with side information at the receiver* [84] (see Figure 2.2). Namely, they gave a rate-distortion function $R_{WZ}^*(D)$ for the problem of encoding one source $X$, guaranteeing an average fidelity of $E\{d(X, \hat{X})\} \leq D$, assuming that the other source $Y$ (playing the role of side information) is available losslessly at the decoder, but not at the encoder. In particular, they showed that, although Wyner-Ziv coding usually suffers rate loss compared to the case where the side information is available at both the encoder and decoder, there is no performance loss if the two correlated sources $X$ and $Y$ are jointly Gaussian and a mean-squared error (MSE) is used as the distortion metric (i.e., $d(X, \hat{X}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$).

In the late 1990s, Zamir [92] showed that the rate loss in the general Wyner-Ziv problem is smaller than 0.5 bit/source symbol compared to the joint encoding and decoding scenario when a squared-error distortion metric is used at the decoder.

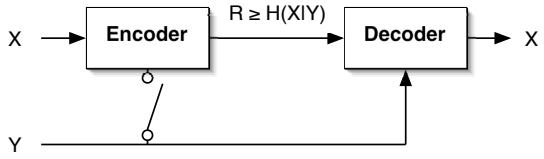


**Figure 2.2: Lossy compression of $X$ with side information $Y$. Wyner and Ziv showed that if $X$ and $Y$ are jointly Gaussian and MSE is used as the distortion metric, there is no performance loss whether the side information $Y$ is available at the encoder or not, as long as it is available at the decoder.**

Zamir and Shamai [93] also showed the theoretical feasibility of a lattice code framework for Wyner-Ziv encoding. For Gaussian sources, a Wyner-Ziv encoder can be seen as a quantizer followed by a Slepian-Wolf encoder. Therefore, the design of specific quantizers for encoding with side information available at the decoder has attracted a significant interest [87, 90, 93].

Slepian-Wolf and Wyner-Ziv coding are source coding problems. However, a strong link to channel coding exists, since the practical binning schemes used at the encoders are usually based on a partitioning of the space using linear channel codes. This relationship between distributed source coding and channel coding was first studied by Wyner in [83]. Further, Csiszár [14] showed that the Slepian-Wolf bound is universally achievable by linear codes using random coding arguments. The next section presents the general idea behind the design of practical coders based on channel coding principles.

## 2.2 Practical Coders

In DISCUS [52], Pradhan and Ramchandran proposed for the first time a practical coding technique for DSC inspired by channel coding techniques. In order to give the correct intuition behind the DISCUS approach, we first present a simple example: Assume $x$ and $y$ are two uniformly distributed 3-bit sequences that are correlated

such that their Hamming distance ($d_H(x, y)$) is at most one (i.e., for any realization of $x$, $y$ is either equal to $x$ or only differs at one bit's position). Therefore, given a certain $x$, we know that the corresponding $y$ belongs to an equiprobable set of four codewords. The following entropies can thus be given: $H(x) = H(y) = 3$ bits, $H(x|y) = H(y|x) = 2$ bits and $H(x, y) = H(x) + H(y|x) = 5$ bits. We know that only 5 bits are therefore necessary to jointly encode $x$ and $y$ losslessly. For instance, one can code $x$ and $y$ jointly by sending one of them completely (3 bits) along with the information representing their difference (2 bits).

Is it now possible to achieve the same coding efficiency when using two independent encoders as in Figure 2.1(b) ? As proven by Slepian-Wolf, the answer is indeed yes! The solution consists in grouping the different codewords into bins. Assume that $y$ is transmitted completely to the decoder (using 3 bits), and consider the following set of bins containing all the possible outcomes for $x$: $\text{bin}_0 = \{000, 111\}$, $\text{bin}_1 = \{001, 110\}$, $\text{bin}_2 = \{010, 101\}$ and $\text{bin}_3 = \{100, 011\}$. Note that the codewords have been placed into the bins such that the Hamming distance between the members of a given bin is maximal (3 in this case). Now, instead of transmitting $x$ perfectly to the decoder (3 bits), only the index of the bin that $x$ belongs to is transmitted (2 bits). On receiving this information, the decoder can retrieve the two possible candidates for $x$. Finally, since their distance to each other is three, only one of them can satisfy the correlation with $y$ given by: $d_H(x, y) \leq 1$. By observing $y$, the decoder can therefore retrieve the original $x$. This is like saying that there is only one $x$ in each bin that is jointly typical with $y$. Therefore, exact reconstruction of $x$ and $y$ is in this case always possible.

This intuitive example can be generalized using linear channel codes. Assume that $x$ and $y$ are two uniformly distributed $n$-bit sequences that are correlated such that their Hamming distance is at most $m$, i.e. $d_H(x, y) \leq m$. Consider an $(n, k)$ binary linear code $\mathcal{C}$, given by its parity check matrix $\mathbf{H}$, that can correct up to

$M \geq m$ errors per $n$-bit codeword [59]. We call *coset number* $i$ the set $\{x_j\}_{j=1}^{2^k}$ of all $n$-bit codewords that have a syndrome equal to $i$, that is, such that $\mathbf{H}x_j^T = i$. Therefore, the code $\mathcal{C}$ generates $2^{n-k}$ cosets having $2^k$ members each. Moreover, any pair of codewords belonging to the same coset have a Hamming distance larger than $2M$. Similarly to our previous example, the distributed coding strategy operates as follows: $y$ is sent perfectly from the second encoder ($n$ bits). The first encoder only transmit the index of the coset given by the syndrome $s_x = \mathbf{H}x^T$ ($n-k$ bits). At the decoder, the original $x$ can be recovered as the only member of coset $s_x$ satisfying the correlation ($d_H(x, y) \leq m$) with the received $y$. This distributed encoding approach thus requires that a total of only $2n - k$ bits be transmitted in order to reconstruct $x$ and $y$ losslessly.

The link between distributed source coding and channel coding (error-correcting coding [59]) is highlighted in Figure 2.3. In channel coding, a redundant codeword $x$ is generated by adding parity bits to the original information block $c$ to be transmitted, such that, after $x$ is sent through the noisy channel, the corrupted output $y$ still contains enough information to perfectly recover $c$. In other terms, the idea is to determine a set of codewords (i.e., a code $\mathcal{C}$), such that, when any of them is sent through the noisy channel, the corrupted version received $y$ remains closer to the original $x$ than to any other member of this set with high probability. An appropriate code is therefore chosen based on the joint distribution $p(x, y)$. In distributed source coding, $x$ and $y$ represent the two correlated sources to be transmitted. Assuming that $y$ has been transmitted to the decoder, only the syndrome $s_x$ of $x$ needs to be transmitted from the first source. At the decoder, the set of all $n$-bit sequences having syndrome $s_x$ is retrieved and the only one satisfying the correlation with $y$ is retrieved as the original $x$.

The correlation between the sources can be seen as a "virtual" dependence channel between the sequences $x$ and $y$. The problem of finding a good code for

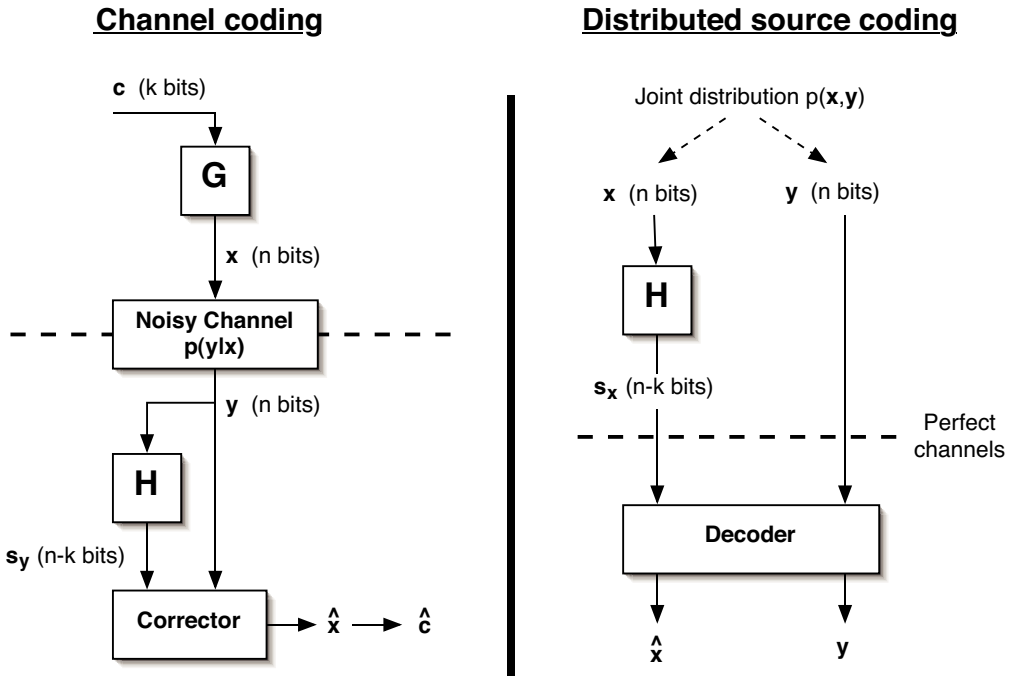


**Figure 2.3: Channel coding vs. Distributed source coding. In channel coding, the syndrome of $y$ is used at the decoder to determine the error pattern. Then the original $x$ is recovered by correcting $y$. In distributed source coding, the syndrome of $x$ is transmitted to the decoder. Knowing $y$ and the syndrome of $x$, the decoder can thus retrieve the difference pattern between $x$ and $y$ and then reconstruct the original $x$.**

distributed source coding is therefore similar to the problem of finding a good channel code for this "virtual" channel.

Practical designs based on advanced channel codes such as Turbo codes [1, 19, 20, 33, 35, 40] and LDPC codes [11, 37, 38, 63, 74, 90] have been proposed in the last few years. All these coding approaches can closely approach the theoretical bounds for different correlation models. However, most of them focus on the asymmetric scenario (corner points of the Slepian-Wolf achievable rate region), also known as compression with side information at the decoder.

For some applications, it may be necessary to have more flexibility in the choice of the rates at which each source should be encoded, rather than being restricted to the corner points. In particular, operating at the mid-point of the Slepian-Wolf rate region (where all the sources are compressed at the same rate) may be desirable in sensor network applications where each sensor node has similar power/communication resources at its disposal (see Section 2.3.1). The most intuitive solution to achieve other operating points than the corners is time-sharing [12]. However, this approach might not be practical as it requires synchronization between the encoders and the decoder, and offers asymptotic disadvantages. Another possible solution is to use source splitting [11, 12, 61], but it has the problem of increasing the complexity and requires a deeper knowledge of the statistics of the "new" subsources.

Pradhan and Ramchandran studied the problem of distributed symmetric encoding of two sources using linear channel codes in [53, 54]. In particular, they showed that the symmetric approaches can achieve exactly the same performance as the asymmetric ones. Practical designs allowing to cover the entire Slepian-Wolf achievable rate region have recently been proposed in [11, 20, 23, 62, 64, 74].

## 2.3 Recent Applications of DSC

### 2.3.1 Wireless Sensor Networks

Sensor networks [4] have been attracting a significant interest in recent years. Advances in wireless communication technologies and hardware have enabled the design of these cheap low-power miniature devices that make up sensor networks. The truly distributed (or decentralized) nature of sensor networks is radically changing the way in which we sense, process and transport signals of interest. In the classical "many-to-one" scenario, many sensors are spatially deployed to monitor some physical phenomenon of interest, and transmit independently their measurements to a common central receiver. The main task of the receiver is then to reproduce the best possible estimation of the observed phenomenon, according to some distortion metric. The sensors can typically be temperature, pressure or humidity sensors, but also microphones or cameras.

Due to the spatial proximity of the sensors and the physical properties of the observed phenomenon, the data acquired among the sensors can be highly correlated. Transmitting this data directly from each sensor to the common receiver could thus imply the communication of a large amount of redundant information. Since the sensors have limited power resources, efficient data transmission is therefore crucial to guarantee the survival of such systems. This particular constraint makes DSC of great interest for wireless sensor network applications.

A first theoretical extension of DSC in a sensor network setting was proposed by Flynn and Gray in [18], where they derived achievable rate-distortion bounds for the case of two sensors transmitting their encoded observations of the phenomenon to a common receiver, and discussed practical ways of designing quantizers to reach good coding performances.

More recently, several studies on decentralized detection and estimation problems have been proposed [5, 50, 86]. The CEO problem [50] provides an interesting theoretical framework for distributed sensing systems in multiuser communications. In this scenario, a hidden data sequence $\{X(t)\}_{t=1}^{\infty}$ is of interest to a central unit or the CEO (Chief Executive Officer). Since the CEO cannot observe this process directly, he (or she) employs $L$ agents (or sensors) to observe independently corrupted versions of the source sequence. The agents then encode independently their observations and transmit the data to the CEO through rate-constrained noiseless channels.

A recent comprehensive coverage of distributed compression-estimation in wireless sensor networks was proposed in [86]. In many of these sensor network applications, the aim of the central receiver is not to fully reconstruct the observed phenomenon, but only to infer some underlying hypotheses [5]. In this context, an interesting question is to determine whether it is more efficient to let the sensors first run a local estimation/detection process and then transmit their results to the main receiver for fusion, or simply let the sensors transmit their acquired data and have the receiver run a joint estimation/detection process. The answer to this question directly depends on the physical properties of the observed phenomenon and the constraints considered (e.g. power constraints, processing capabilities, bandwidth efficiency, scalability, robustness to changes in network or environment).

DSC plays a particularly important role in dense wireless sensor networks where the number of sensors can be very large. Several works focusing on the scalability properties of sensor networks have appeared recently [28, 31, 45, 46, 49]. The main issue is to understand if the use of DSC can be sufficient to compensate for the increase of data transmission as the number of sensors grows. Several authors have presented negative results [45, 46] arguing that if the total amount of data that can be received by the central decoder is limited, then the global performance of the

network goes to zero as the number of sensors grows to infinity. More positive results were then proposed in [31, 49] where an upper bound for the global rate-distortion behaviour was given.

A particularly interesting scenario arises when the observed phenomenon can be represented using a parametric model with a finite number of parameters, or has a finite rate of innovation (FRI [80]). This property means that it is possible to reconstruct perfectly the phenomenon using a finite number of sensors that spatially sample it. In this case, it was shown [31] that it is possible to efficiently trade-off the density of sensors with the sensing resolution at each sensor, when the number of sensors increases beyond the critical sampling, thanks to distributed source coding approaches.

### 2.3.2 Distributed Video Coding

In video coding standards such as MPEG-x or H.26x [47], the encoder usually tries to exploit the statistics of the source signal in order to remove, not only spatial, but also temporal redundancies. This is usually achieved by using motion-compensated predictive encoding, where each video frame is encoded using a prediction, based on previously encoded frames, as side information. This side information (the predictor) must therefore be available at both the encoder and the decoder in this case.

The idea of distributed video coding is to employ DSC approaches in order to allow for an independent encoding of the different frames at the encoder, while letting to the decoder the burden of exploiting the temporal dependencies. In other terms, each video frame is encoded independently knowing that some side information will be available at the decoder (the side information can typically be a prediction based on previously decoded frames).

The first very interesting aspect of distributed video coding is that it con-

siderably reduces the complexity of the video encoder by shifting all the complex interframe processing tasks to the decoder. This property can be of great interest for power/processing limited systems such as wireless camera sensors that have to compress and send video to a fixed base station in a power-efficient way. Here, it is assumed that the receiver has the ability to run a more complex decoder. In the case where the receiver of the compressed video signal is another complexity-constrained device, a solution using a more powerful video transcoder somewhere on the network can be used (see Figure 2.4).

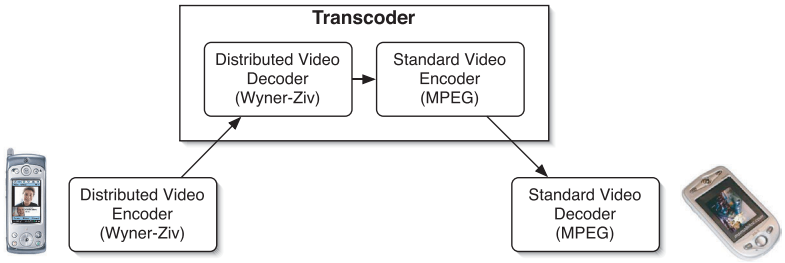


**Figure 2.4: Transcoding architecture for wireless video. This method allows for low-complexity encoder (Wyner-Ziv encoder) and decoder (MPEG-like decoder) at both wireless devices. However, this architecture relies on the use of a complex transcoder somewhere on the network.**

Another strong advantage of distributed video coding is that it is naturally robust to the problem of drift between encoder and decoder. The drift problem is due to prediction mismatch that can happen due to channel losses and usually creates visual artefacts that propagates until the next intra-coded frame is received. This built-in robustness is due to the fact that the encoding is not based on a specific prediction, but only assumes that a relatively good predictor will be available at the decoder. Therefore, slightly different predictors can lead to a correct decoding. This particular property highlights the fact that Wyner-Ziv coding can actually be seen as a joint source-channel coding approach (see Section 2.3.4).

The first video coding approach based on distributed compression principles

was proposed in [57], and is known as *PRISM* or "Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding". We recommend the reader to refer to this original work to obtain more information about their specific coding architecture. Other approaches have since been proposed [17, 24, 56, 66, 89] and the area of distributed video coding is currently gaining a fast growing interest.

Although all these approaches are extremely promising, they are still not as efficient as standard video coders in terms of rate-distortion performance. The gap is mainly due to the fact that distributed source coding techniques usually rely on the fact that the correlation structure is known a-priori. It is therefore only with the knowledge of this correlation that optimal codes can be designed. The estimation of this correlation has proven, however, to be extremely difficult.

Another reason for the performance gap between theory and practice is that the decoder has to search for the best side information (corresponding to the reference information in predictive coding) in previously decoded data. The side information extracted and used by the decoder is typically not as good (in the sense of being as close as possible to the data to be encoded) as the reference information used in predictive coding systems.

### 2.3.3 Distributed Multi-View Coding

Compression techniques for multi-view images have attracted a deep interest during the last decade. This is partly due to the introduction of several new $3D$ rendering techniques such as image-based rendering (IBR) [71] and lightfield rendering (LFR) [34] that represent real-world $3D$ scenes using a set of images obtained from fixed viewpoint cameras. The amount of raw data acquired by practical systems can be extraordinary large and typically consists of hundreds of different views. Due to the spatial proximity of the different cameras, an extremely large amount of redun-

dant information is present in the acquired data. Compression is therefore highly needed.

In order to exploit the correlation between the different views, a joint encoder could be employed. However, this would require that all the cameras first transmit their data to a common receiver that would have to store it and then perform the joint compression. This would clearly use a tremendous amount of communication resources and storage space, and might not be feasible in some practical settings. For these reasons, it would be preferable to compress the images directly at the cameras using distributed compression techniques. The main advantages of such an approach is that it would only require a low-complexity encoder at each camera, and would considerably reduce the overall amount of transmission necessary from the cameras to the central decoder. Moreover, the compressed data could be directly stored at the receiver using optimal memory space. Nevertheless, in this case the decoder is assumed to be more sophisticated in order to handle the high-complexity joint decoding of the views.

Several approaches for distributed multi-view image coding directly inspired from distributed video coding techniques have been proposed [2, 30, 36, 77–79, 94]. The basic idea is to see each different view as a frame of a video sequence and apply a Wyner-Ziv video coding approach to them. Nevertheless, these approaches suffer from several drawbacks: First, they require that some cameras transmit their full information (to provide side information to the receiver) while others only transmit partial information. This makes them clearly asymmetric, which can be a problem for some practical applications. Second, while the correlation between successive video frames can be difficult to estimate, basic multi-view geometry could be used when dealing with multi-camera systems. However, most of these approaches do not take advantage of this information so as to improve the performance of their encoders. In [43], Maitre et al. propose to exploit the scene geometry to design

enhanced interpolation approaches for the problem of side information generation for distributed video coding of static scenes. Their results show substantial performance gains over classical approaches using block-based motion-compensated frame interpolation.

Another distributed multi-view compression approach based on correspondence analysis and super-resolution has been proposed in [81], but it requires that some information is transmitted between the encoders to perform some coarse registration of the different views. This approach is therefore not fully distributed and cannot be used for all the applications where communication between the encoders is impossible.

Distributed compression approaches focusing on the encoding of multiple blurred and noisy images of a scene acquired from the same viewing position have been proposed in [21, 39]. Although their results show performances that are relatively close to the theoretical bounds, their approaches cannot be used for multi-view images, because of the limitation of their correlation model.

Finally, extensions of distributed video coding to distributed multi-view video coding have been proposed in [26, 73, 82, 91]. In this scenario, the decoder has to exploit the redundancy between all the received video signals (temporal and view correlations) to make the best possible prediction and generate the side information.

In Chapter 3, we propose a distributed coding strategy for camera sensor networks that uses some geometrical information about the scene and the position of the cameras in order to estimate precisely the correlation in the visual information.

### 2.3.4 Joint Source-Channel Coding

As stated in a comprehensive review on distributed source coding proposed by Xiong et al. [88] ; “Wyner-Ziv coding is, in a nutshell, a source-channel coding problem”.

This property of Wyner-Ziv coding was emphasized in Section 2.3.2, where we highlighted the fact that distributed video coding presents a natural robustness to the problem of drift [24, 57, 66, 89]. In fact, Wyner-Ziv coding can be thought of as a channel coding technique that is used to correct the "errors" between the source to be coded and the side information available at the decoder. We can thus see the relationship between the source and the side information as represented by a "virtual" correlation channel. Then, if a good channel code for this virtual channel can be found, it would clearly provide us with a good Wyner-Ziv code through the associated coset codes.

In case of transmission over a non-perfect channel, it seems quite intuitive that the use of a stronger Wyner-Ziv code could not only compensate for the discrepancies between the source and the side information, but also correct errors due to the unreliable transmission of the source sequence. Several papers addressing this particular property of distributed source coding have been published [41, 65]. A recent approach (based on PRISM) that allows for an error-resilient distributed compression of multi-view video sequences was proposed in [91].

Finally, Wyner-Ziv coding is also strongly related to systematic lossy source-channel coding [67], where an encoded version of the source signal is sent over a digital channel to serve as enhancement information to a noisy version of the source signal received through an analog channel. Here, the noisy (analog) version of the source signal plays the role of side information for decoding the information received from the digital channel. A detailed description of video coding based on systematic lossy source-channel coding can be found in [58].

## 2.4 Conclusions

In this chapter, we have presented the theoretical foundations of distributed source coding and highlighted the practical coders recently proposed in the literature. We have also discussed some of the recent applications of DSC such as distributed video coding or wireless sensor networks.

# Chapter 3

# The Plenoptic Function and its Distributed Compression: A preliminary analysis

## 3.1 Introduction

The aim of this chapter is to present a set of preliminary results that will then be used to derive fundamental performance bounds in camera sensor networks (Chapter 4), to develop a new distributed image compression algorithm (Chapter 5) and for distributed compression of binary sources (Chapter 6).

Distributed compression schemes usually rely on the assumption that the correlation of the source is known a-priori. In this chapter, we show how it is possible to estimate the correlation structure in the visual information acquired by a multi-camera system by using some simple geometrical constraints, and present a coding approach that can exploit this correlation in order to reduce the total amount of information to be transmitted from the visual sensors to the common central receiver. The coding scheme we propose allows for a flexible distribution of the bit-rates

amongst the encoders and is optimal in many cases. Our technique can also be made resilient to a fixed number of visual occlusions, when certain objects of the scene are only visible from a fraction of the camera positions.

## 3.2 The Plenoptic Function

The plenoptic function was first introduced by Adelson and Bergen in 1991 [3]. It corresponds to the function representing the intensity and chromaticity of the light observed from every position and direction in the 3D space, and can therefore be parameterized as a 7D function: $I_{PF} = P(\theta, \phi, \omega, \tau, V_x, V_y, V_z)$. The three coordinates $(V_x, V_y, V_z)$ correspond to the position of the camera, $\theta$ and $\phi$ give its orientation, $\tau$ is the time and $\omega$ corresponds to the frequency considered. The measured parameter $I_{PF}$ is simply the intensity of the light observed under these parameters. The plenoptic function represents thus the visual information available from any viewing position around a scene of interest. Hence, image-based rendering (IBR) techniques can be thought of as methods that try to reconstruct the continuous plenoptic function from a finite set of views [71]. Once the plenoptic function has been reconstructed, it is then straightforward to generate any view of the scene by setting the appropriate parameters. The high dimensionality of this function makes it, however, extremely impractical. By fixing the time $\tau$ and the frequency $\omega$ (or integrating over the range of wavelengths considered), and assuming that the whole scene of interest is contained in a convex hull (and observed from the outside of this hull), the plenoptic function can be reduced to a 4-D function [25, 34]. The parameterization of this 4-D function is usually done using two parallel planes: the focal plane (or camera plane) and the retinal plane (or image plane). A ray of light is therefore parameterized by its intersection with these two planes as shown in Figure 3.1(a). The coordinates in the focal plane $(s, t)$ give the position of the

pinhole camera, while the coordinates in the retinal plane $(u, v)$ give the point in the corresponding image.

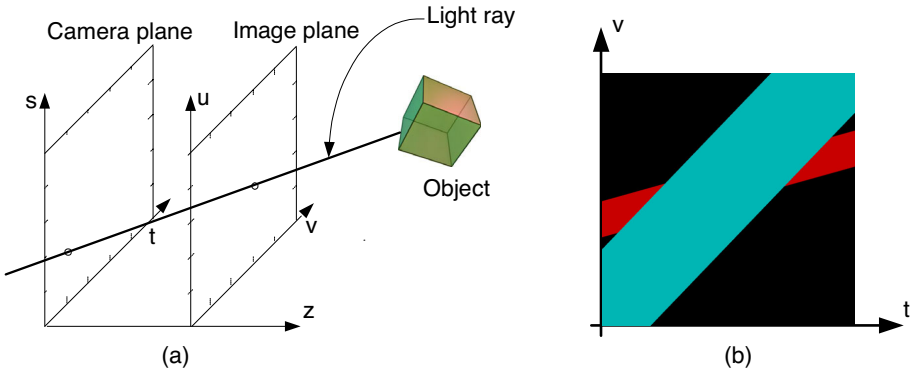


**Figure 3.1: (a) 4-D lightfield parameterization model where each light ray is characterized by its intersections with the camera and focal planes. (b) Epipolar ($v$-$t$) plane image of a scene with two objects at different depths (the $s$ and $u$ coordinates are fixed in this case).**

An *Epipolar Plane Image (EPI)* represents a 2-D subspace of the plenoptic function. It can be used to represent and analyse the redundancy in the multi-view data. For example, the $(v, t)$ plane is typically used to represent the epipolar geometry of a scene when the different cameras are placed on a horizontal line (see Figures 3.1(b) and 3.2).

### Acquisition of the plenoptic function: some examples from 2-D to 7-D

- **2-D**: The EPI shown in Figure 3.1(b) corresponds to a 2-D slice of the plenoptic function (see also Figure 3.2). It is obtained by moving a camera along a horizontal line and by acquiring a horizontal scan-line from each camera position. In another scenario, the cameras could be placed on a circle (oriented towards the outside of the circle and looking perpendicular to it). Notice that a point of the scene would not follow a straight line any more with this 2-D representation of the plenoptic function.

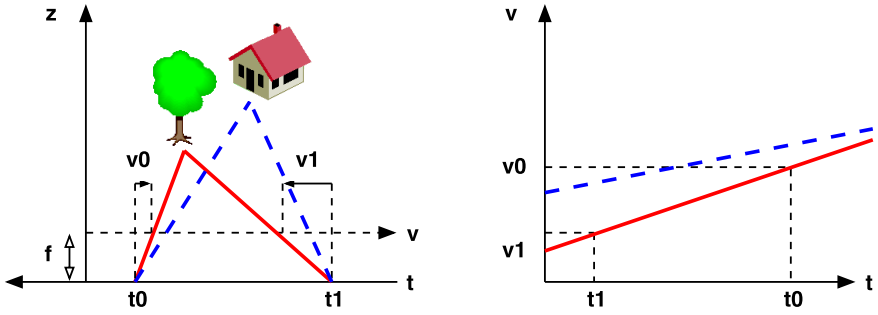


**Figure 3.2: 2D plenoptic function of two points of a scene. The $t$-axis corresponds to the camera position and the $v$-axis corresponds to the relative positions on the corresponding image. A point of the scene is therefore represented by a line whose slope is directly related to the point's depth ($z$-axis). The difference between the positions of a given point on two different images thus satisfies the relation $(v - v') = \frac{f(t-t')}{z}$, where $z$ is the point's depth and $f$ is the focal length of the cameras.**

- **3-D**: We consider now that the cameras are still on a line and that a full 2-D image is acquired from each camera position. The 3-D data obtained has a structure that is similar to a video sequence, but the motion of the objects can be fully characterized by their positions in the scene.

- **4-D**: The *lightfield* [34] and the *lumigraph* [25] are 4-D representations of the plenoptic function where the cameras are placed on a 2-D plane and full 2-D images are acquired for all the camera positions. The cameras could also be placed on the surface of a sphere, looking towards the centre of the sphere, assuming that the whole scene of interest is contained in the sphere.

- **5-D**: Allowing the cameras to be placed anywhere in the 3-D space, we now obtain a 5-D representation of the plenoptic function. In a similar manner, considering the evolution of time with the acquisition of a lightfield, we also move from a 4-D to a 5-D representation.

- **6-D**: If there is no restriction on the position of the cameras and the 2-D views are acquired over time, the resulting data can be represented as a 6-D

function.

- **7-D**: In the previous examples we assumed that we were restricted to one (grayscale) or three (RGB) frequencies for the observed light rays. In order to represent the intensity of the light for any frequency, from any position and direction, and at any time, the full 7-D plenoptic function is required.

## 3.3 The Linear Camera Sensor Network Configuration

A camera sensor network is able to acquire a finite number of different views of a scene at any given time and can thus be seen as a sampling device for the plenoptic function. We choose the following scenario for our work: Assume that we have $N$ cameras placed on a horizontal line. Let $\alpha$ be the distance between two consecutive cameras, and assume that they are all looking in the same direction (perpendicular to the line of cameras). Assume that the observed scene is composed of simple objects such as uniformly colored polygons parallel to the image plane and with depths bounded between the two values $z_{min}$ and $z_{max}$ as shown in Figure 3.3. According to the epipolar geometry principles, which are directly related to the structure of the plenoptic function (see Figure 3.2), we know that the difference between the positions of a specific object on the images obtained from two consecutive cameras will be equal to $\Delta = \frac{\alpha f}{z}$, where $z$ is the depth of the object and $f$ is the focal length of the cameras. This disparity $\Delta$ depends only on the distance $z$ of the point from the focal plane. If we know a-priori that there is a finite depth of field, that is $z \in [z_{min}, z_{max}]$, then there is a finite range of disparities to be coded, irrespective of how complicated the scene is. This key insight can be used to develop new distributed compression algorithms as we show in the next section.

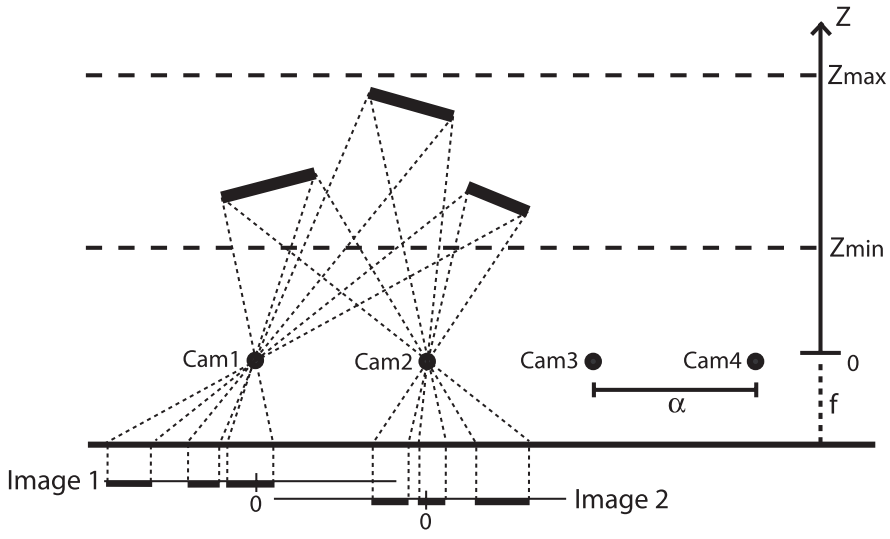


**Figure 3.3: Our camera sensor network configuration.**

Notice that a similar insight has been previously used by Chai et al. to develop new schemes to sample the plenoptic function [9].

## 3.4 A Novel Distributed Coding Approach for Simple Scenes

In this section, we propose a lossless distributed coding scheme for the configuration presented in Figure 3.3 in the case where there are only two cameras. Since both encoders have some knowledge about the geometry of the scene, the correlation structure of the two sources can be easily retrieved. We then show that our coding technique can be used with any pair of bit-rates contained in the achievable rate region defined by Slepian and Wolf [72], and can therefore be optimal.

### 3.4.1 Distributed stereo coding

Let $X$ and $Y$ be the horizontal discrete positions (at unit pixel precision) of a specific object on the images obtained from two consecutive cameras. Assume that the images are squared images with $2^{2R}$ pixels each (The width and height of an image is $2^R$ pixels long). Due to the epipolar geometry and the information we have about the scene, that is $(\alpha, f, z_{min}, z_{max})$, we know that $Y \in [X+\frac{\alpha f}{z_{max}}, X+\frac{\alpha f}{z_{min}}]$ for a specific $X$. Encoding $X$ and $Y$ independently would require a total of $H(X)+H(Y)$ bits. However, using a *coset* approach, we can transmit $X$ losslessly and modulo encode $Y$ as $Y' = Y \bmod \lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$. Here, the modulo of $Y$ can be seen as the syndrome of $Y$ as it corresponds to the minimum information required in order to perfectly reconstruct $Y$, knowing $X$ and the correlation structure. In binary linear codes, a coset is constructed by gathering codewords such that a maximum Hamming distance is kept between any pair of codewords in the coset. In our case, we consider Euclidean distance and we want to make sure that the difference between any pair of values in a given coset is not smaller than $\lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$. Therefore, all the values contained in any range of size $\lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$ must belong to different cosets, which can be done by using the modulo operator.

By observing $X$ and $Y'$, the receiver will then retrieve the correct $Y$ such that $Y \in [X + \frac{\alpha f}{z_{max}}, X + \frac{\alpha f}{z_{min}}]$. The overall transmission rate is therefore decreased to $H(X) + H(Y')$ bits. If we assume that the difference between $X$ and $Y$ is uniformly distributed in $[\frac{\alpha f}{z_{max}}, \frac{\alpha f}{z_{min}}]$, we can claim that $H(Y') = H(Y|X)$ (we know that $Y'$ is uniformly distributed over a set of values of the same size as the number of possible disparities between $X$ and $Y$). We can therefore see that our coding scheme using $H(X) + H(Y') = H(X) + H(Y|X) = H(X, Y)$ bits is optimal.

This simple distributed coding technique is powerful since it takes full advantage of the geometrical information to minimize the global transmission bit-rate.

However, its asymmetric repartition of the bit-rates may be problematic in some practical applications. In the following, we will show that this coding approach can be extended in a way such that any pair of bit-rates satisfying the Slepian and Wolf conditions can be used.

Looking at the following relation: $H(X,Y) = H(X|Y) + H(Y|X) + I(X,Y)$, we can see that the minimum information that must be sent from the source $X$ corresponds to the conditional entropy $H(X|Y)$. Similarly, the information corresponding to $H(Y|X)$ must be sent from the source $Y$. The remaining information required at the receiver in order to recover the values of $X$ and $Y$ perfectly is related to the mutual information $I(X,Y)$ and is by definition available at both sources. This information can therefore be obtained partially from both sources in order to balance the transmission rates.

We know that the correlation structure between the two sources is such that $Y$ belongs to $[X + \frac{\alpha f}{z_{max}}, X + \frac{\alpha f}{z_{min}}]$ for a given $X$. Let $\widetilde{Y}$ be defined as $\widetilde{Y} = Y - \lceil \frac{\alpha f}{z_{max}} \rceil$. This implies that the difference $(\widetilde{Y} - X)$ is contained in $\{0, 1, \ldots, \delta\}$, where $\delta = \lceil \alpha f (\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$. Looking at the binary representations of $X$ and $\widetilde{Y}$, we can say that the difference between them can be computed using only their last $R_{min}$ bits where $R_{min} = \lceil \log_2(\delta + 1) \rceil$. Let $X_1$ and $\widetilde{Y}_1$ correspond to the last $R_{min}$ bits of $X$ and $\widetilde{Y}$ respectively. Let $X_2 = (X \gg R_{min})$ and $\widetilde{Y}_2 = (\widetilde{Y} \gg R_{min})$, where the "$\gg$" operator corresponds to a binary shift to the right. We can thus say that $\widetilde{Y}_2 = X_2$ if $\widetilde{Y}_1 \geq X_1$ and that $\widetilde{Y}_2 = X_2 + 1$ if $\widetilde{Y}_1 < X_1$. As presented in Figure 3.4, our coding strategy consists in sending $X_1$ and $\widetilde{Y}_1$ from the sources $X$ and $Y$ respectively and then, sending only a subset of the bits for $X_2$ and only the complementary one for $\widetilde{Y}_2$. At the receiver, $X_1$ and $\widetilde{Y}_1$ are then compared to determine if $\widetilde{Y}_2 = X_2$ or if $\widetilde{Y}_2 = X_2 + 1$. Knowing this relation and their partial binary representations, the decoder can now perfectly recover the values of $X$ and $\widetilde{Y}$.

Assume that $z_{min}$ and $z_{max}$ are such that $(\delta + 1)$ is a power of 2. If we

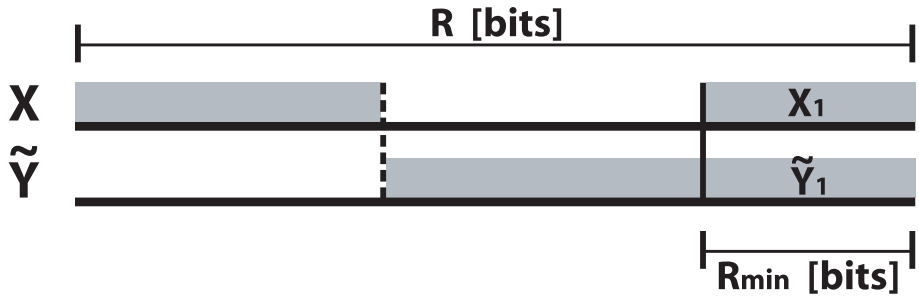


**Figure 3.4: Binary representation of the two correlated sources. The last $R_{min}$ bits are sent from the two sources but only complementary subsets of the first $(R - R_{min})$ bits are necessary at the receiver for a perfect reconstruction of $X$ and $Y$.**

assume that $(\widetilde{Y} - X)$ is uniformly distributed, we can state that $H(\widetilde{Y} - X) = H(X|Y) = H(Y|X) = R_{min}$. Let $S(X_2)$ be a subset of the $R - R_{min}$ bits of $X_2$ and let $\bar{S}(\widetilde{Y}_2)$ corresponds to the complementary subset of $\widetilde{Y}_2$. If we assume now that $X$ is uniformly distributed in $\{0, 1, \ldots, 2^R - 1\}$, we have that $H(S(X_2)) + H(\bar{S}(\widetilde{Y}_2)) = H(S(X_2), \bar{S}(\widetilde{Y}_2)) = I(X, Y)$. The total rate necessary for our scheme corresponds to $I(X, Y) + 2R_{min} = H(X, Y)$ and is therefore optimal. We can now summarize our results into the following proposition:

**Proposition 3.1.** *Consider the configuration presented in Figure 3.3 with two cameras, and assume that no occlusion occurs in the two corresponding views. The following distributed coding strategy is sufficient to allow for a perfect reconstruction of these two views at the decoder. For each object's position:*

- *Send the last $R_{min}$ bits from both sources, with $R_{min} = \lceil \log_2(\delta + 1) \rceil$ and $\delta = \lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$.*
- *Send complementary subsets for the first $(R - R_{min})$ bits.*

*If we assume that $X$ and $(Y - X)$ are uniformly distributed and that $\delta = 2^{R_{min}} - 1$, this coding strategy achieves the Slepian-Wolf bounds and is therefore optimal.*

### 3.4.2 The problem of occlusions

In order to reconstruct the position of an object on an image obtained from any virtual camera position, we need to know its correct position in at least two different views. Using the epipolar geometry principles, we can then easily retrieve its absolute position and depth. Unfortunately, a specific object may not be visible from certain viewing positions since it might be hidden behind another object or might be out of field. Nevertheless, using a configuration with more cameras will make it more likely for any object to be visible in at least two views.

Assume we have three cameras in a configuration similar to the one presented in Figure 3.3 and that each object of the scene can be occluded in at most one of these three views. Our goal is to design a distributed coding scheme for these three correlated sources such that the information provided by any pair of these sources about the position of any object of the scene is sufficient to allow for a perfect reconstruction at the receiver. Let $X$, $Y$ and $Z$ be the horizontal positions of a specific object on the images obtained from camera 1, 2 and 3 respectively. We know that $Y$ belongs to $[X+\frac{\alpha f}{z_{max}}, X+\frac{\alpha f}{z_{min}}]$ and $Z$ belongs to $[X+2\frac{\alpha f}{z_{max}}, X+2\frac{\alpha f}{z_{min}}]$ for a given $X$. Moreover, we know that any of these variables is deterministic given the two others and follows the relation $Z = 2Y - X$. Let $\widetilde{X}$ and $\widetilde{Z}$ be defined as $\widetilde{X} = X + \frac{\alpha f}{z_{mean}}$ and $\widetilde{Z} = Z - \frac{\alpha f}{z_{mean}}$ where $z_{mean}$ is given by $\frac{1}{z_{mean}} = \frac{1}{2}(\frac{1}{z_{min}} + \frac{1}{z_{max}})$. It implies that the differences $(Y - \widetilde{X})$ and $(\widetilde{Z} - Y)$ are equal and are included in $[-\delta/2, \delta/2]$ and that the difference $(\widetilde{Z} - \widetilde{X})$ is included in $[-\delta, \delta]$, where $\delta$ is defined as in Section 3.4.1.

Looking at the binary representation of $\widetilde{X}$, $Y$ and $\widetilde{Z}$ (at unit pixel precision), we can say that the difference between any pair can be retrieved using only their last $R_{min}$ bits, where $R_{min} = \lceil \log_2(2\delta + 1) \rceil$. Let $\widetilde{X}_1$, $Y_1$ and $\widetilde{Z}_1$ correspond to the last $R_{min}$ bits of $\widetilde{X}$, $Y$ and $\widetilde{Z}$ respectively. Using a similar approach to that presented in

Section 3.4.1, we know that any complementary binary subsets of $\widetilde{X}_2$, $Y_2$ and $\widetilde{Z}_2$ are necessary at the receiver to allow for a perfect reconstruction. Since one occlusion can happen, we have to choose the binary subsets such that any pair of these subsets contains at least one value for each of the $(R - R_{min})$ bits. A possible repartition is shown in Figure 3.5 (symmetric case). A transmission rate of $\frac{2}{3}r + R_{min}$ for each source is necessary in this case, where $r = R - R_{min}$.

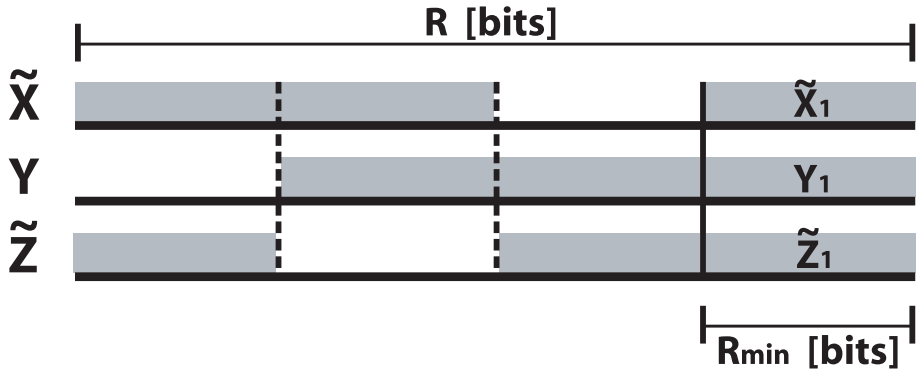


**Figure 3.5: Binary representation of the three correlated sources. The last $R_{min}$ bits are sent from the three sources but only subsets of the first $(R-R_{min})$ bits are necessary at the receiver for a perfect reconstruction of $X$, $Y$ and $Z$ even if one occlusion occurs.**

On receiving the last $R_{min}$ bits from only two sources, the decoder is able to retrieve the last $R_{min}$ bits of the third one, which may be occluded. Therefore, the relationship between $\widetilde{X}_2, Y_2$ and $\widetilde{Z}_2$ can be obtained and only subsets of their binary representations are necessary for a perfect reconstruction. Since an occlusion may have occurred, each bit position has been sent from two different sources, which implies a global transmission of $2r$ bits for the first $(R - R_{min})$ bits. It is therefore apparent that our total transmission rate of $\frac{2}{3}r + R_{min}$ bits per sources is optimal.

### 3.4.3 $N$ cameras with $M$ possible occlusions

We can now generalize our result to any number of cameras and occlusions with the following proposition (see Figure 3.6):

**Proposition 3.2.** *Consider a system with $N$ cameras as depicted in Figure 3.3. Assume that any object of the scene can be occluded in at most $M \leq N-2$ views. The following distributed coding strategy is sufficient to allow for a perfect reconstruction of these $N$ views at the decoder and to interpolate any new view:*

- *Send the last $R_{min}$ bits of the objects' positions from only the first $(M+2)$ sources, with $R_{min} = \lceil \log_2((M+1)\delta) \rceil$ and $\delta = \lceil \alpha f(\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$.*
- *For each of the $N$ sources, send only a subset of its first $(R - R_{min})$ bits such that each particular bit position is sent from exactly $(M+1)$ sources.*

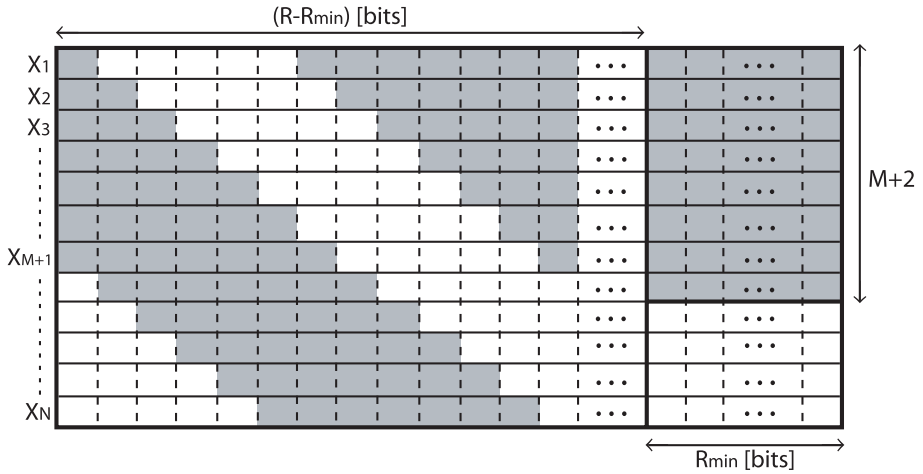


**Figure 3.6: Binary representation of the $N$ correlated sources. The last $R_{min}$ bits are sent only from the $(M+2)$ first sources. Only subsets of the first $(R - R_{min})$ bits are sent from each source, such that each bit position is sent exactly from $(M+1)$ sources.**

Notice that this distributed coding strategy can be extended to the general problem of DSC of correlated binary sources. A general solution based on linear channel codes and syndrome encoding is presented in Chapter 6.

### 3.4.4 Simulation results

We developed a simulation to illustrate the performance of our distributed compression scheme. We created an artificial scene composed of simple objects such as polygons of different intensities placed at different depths. Our system could then generate any view of that scene for any specified camera position. In the example presented in Figure 3.7, we generated three views of a simple scene composed of three objects such that one of them is occluded in the second view, and another one is out of field in the third view. The three generated images have a resolution of

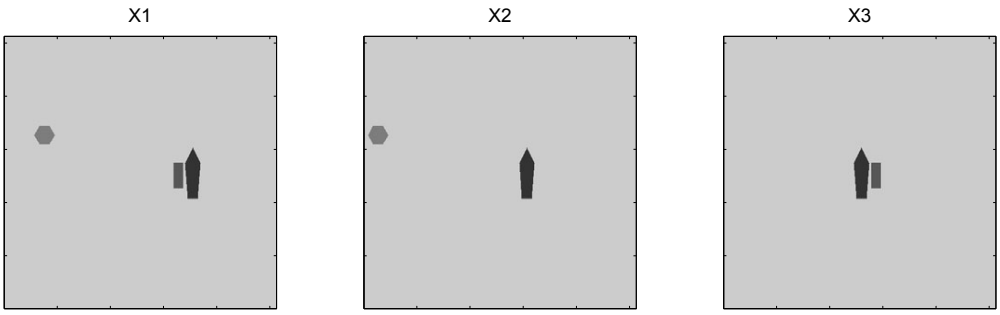


**Figure 3.7: Three views of a simple synthetic scene obtained from three aligned and evenly spaced cameras. Note that an occlusion happens in $X_2$ and that an object is out of field in $X_3$.**

$512 \times 512$ pixels and are used as the inputs for the testing of our distributed compression algorithm. Each encoder applies first a simple corner detection to retrieve the vertex positions of their visible polygons. Each vertex $(x_i, y_i)$ is represented using $2R = 2\log_2(512) = 18$ bits. Each encoder knows the relative locations of the two other cameras ($\alpha = 100$) but does not know the location of the objects on the other images. It only knows that the depths of the objects are contained in $[z_{min}, z_{max}] = [1.95, 5.05]$ and that $f = 1$. Depending on its depth, an object will thus be shifted from 20 to 51 pixels between two consecutive views. This means that the difference $\Delta$ on two consecutive positions (i.e., the disparity) can be described using $R_{min} = \log_2(51 - 19) = 5$ bits.

In order to be resilient to one occlusion, we applied the approach proposed in Section 3.4.2. The results showed that only 14 bits per vertex were necessary from each source (instead of 18) to allow for a perfect reconstruction of the scene at the receiver. When repeating the operation with three other views and assuming that no occlusion was possible, only 8 bits per vertex were necessary from each source.

## 3.5 Conclusions

In this chapter, we have proposed a new lossless distributed compression scheme for camera sensor networks. In particular, we have shown that our approach can be used to encode the positions of the objects in a distributed manner. The method uses some geometrical information about the scene in order to estimate the plenoptic constraints and retrieve a correlation structure for the sources. The encoding process is simple yet very powerful since it can mimic the Slepian and Wolf [72] theoretical performance. A solution to the problem of visual occlusions has also been proposed. In Chapter 5, we will show how these results can be used to develop a practical distributed encoder for real multi-view images.

# Chapter 4

# Fundamental Trade-offs in Camera Sensor Networks: An exact bit-conservation principle

## 4.1 Introduction

In the previous chapter, we presented a distributed compression strategy for multi-view images, where the correlation between the objects positions on the different views can be exploited without requiring any collaboration between the encoders. Our proposed scheme can be used to compress the visual information acquired by a camera sensor network, where each sensor independently processes its data before transmitting the compressed information to a common central receiver. In this chapter, we consider this "many-to-one" sensor network scenario, and study the fundamental trade-offs between the number of sensors and the compression rate used at each encoder in order to guarantee the best global rate-distortion behaviour for the reconstructed views.

Sampling in space and distributed coding are two critical issues in sensor net-

works. For example, deploying too few sensors would lead to a highly aliased reconstruction of the observed phenomenon, while an excessive number of sensors would use all the communication resources by transmitting highly correlated measurements to the receiver. This last issue can actually be addressed by means of distributed source coding techniques (see Chapter 2). The correlated measurements can then be encoded independently at each sensor with a compression performance similar to the one that would be achieved by using a joint encoder.

Despite this powerful coding approach, several authors have presented pessimistic results regarding the scalability of sensor networks [45, 46]. Their main argument comes from the fact that, if the total amount of data that can be received by the central decoder is limited, then the throughput at each sensor scales as $\Theta\left(\frac{1}{N}\right)$ with the number of sensors $N$. The global performance of the network then goes to zero as $N \to \infty$. More optimistic results were recently proposed in [31, 49], where it was shown that, for a given distortion, an upper-bound (independent of $N$) on the total information rate can be given. Recent works have also shown that if the observed phenomenon can be represented with a finite number of degrees of freedom, it is then possible to efficiently trade-off the density of sensors with the sensing resolution at each sensor [28,29] (Notice that the approach presented in [29] requires some communication between neighbouring sensors).

In the context of camera sensor networks, the physical phenomenon that has to be transmitted to the receiver is the visual information coming from the scene (or its plenoptic function [3]), and the samples are the different sampled 2-D views acquired by the sensors. The issues of sampling and communication are particularly interesting in this scenario. Several multi-view sampling theorems have been proposed to address the question of determining the critical sampling of a visual scene under different model assumptions, such as bandlimited scenes [9] or scenes with finite rate of innovation [10]. Starting from this critical sampling, our main

objective is to show how we can arbitrarily increase the number of sensors, while maintaining a constant global rate-distortion behaviour for the reconstruction of the scene at the receiver. This "bit-conservation principle" is achieved by means of a distributed compression scheme for multi-view images based on the approach proposed in Chapter 3.

We consider the camera sensor network set-up proposed in Figure 3.3, and assume that $z_{max}$ can be equal to infinity. The observed scene is made of $L$ Lambertian planar polygons that can be horizontally tilted and placed at different depths. Each of these polygons has a certain polynomial intensity (the intensity along the horizontal and vertical axes varies as a polynomial of maximum degree $Q$). The perspective projection observed at each camera is therefore given by a 2-D piecewise polynomial function. The difference between the $N$ views is that the pieces are shifted differently according to their depths. Moreover, pieces can be linearly contracted or dilated if they correspond to a tilted object, as shown in Figure 4.1.

Since the cameras are placed on a horizontal line, only the horizontal parallax has an effect on the correlation between the $N$ different views. We can therefore reduce the sampling and compression problems to the 1-D case without loss of generality. Throughout this chapter, we focus on one particular horizontal scanline for the different views for the sake of simplicity, but all the results can be easily extended to the 2-D case. Figure 4.1 shows an example of two correlated 1-D views with three pieces of constant intensities placed at different depths.

The $N$ cameras then have to communicate their acquired and processed data to a central station through a multi-access channel with fixed capacity $C$. The natural questions we want to address in this chapter are the following: a) Is there a sampling result that guarantees that perfect reconstruction of the visual scene is possible from a finite number of blurred and sampled projections? b) Since the observed projections have to be transmitted through a channel with fixed capacity, is

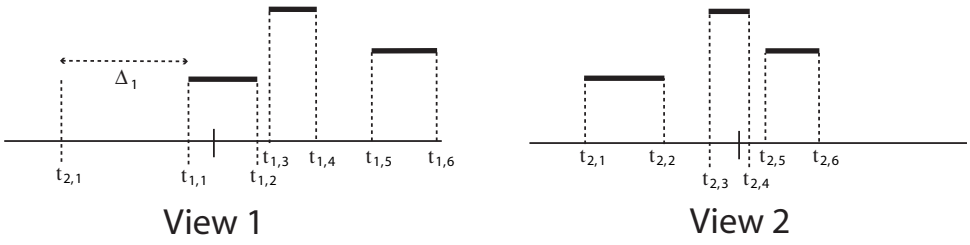


**Figure 4.1: Two correlated views of the same scene observed from two different viewing positions. Each discontinuity is shifted according to the epipolar geometry. The set of disparities is given by $\{\Delta_i\}_{i=1}^{2L}$, where $\Delta_i = t_{1,i} - t_{2,i}$.**

the number of cameras going to influence the reconstruction fidelity at the decoder?

We show in the next sections that an exact sampling theorem for this scenario exists and, most important, we show that a distributed coding strategy inspired by our results presented in Chapter 3 allows for a reconstruction of the scene with a distortion that is independent of the number of sensors and depends only on the total number of bits that are transmitted through the channel.

## 4.2 Distributed Acquisition of Scenes with Finite Rate of Innovation

The signals observed at the sensors are piecewise polynomial signals and can be classified as signals with *Finite Rate of Innovation (FRI)*. Recently, new sampling methods for these classes of non-bandlimited signals have been proposed [15, 80]. They allow for a perfect reconstruction using only a finite number of samples. The sampling can be done using sinc or Gaussian kernels, or any function that can reproduce polynomials. In other terms, each sensor observes a blurred and sampled version of the original piecewise polynomial projection, and is able to reconstruct exactly the original parameters of the view (exact discontinuity locations and polynomial coefficients). Extensions of this sampling approach for 2-D signals with FRI

have been proposed recently [44, 68].

Since each sensor is able to retrieve precisely its original perspective projection, we can show that a finite number of sensors is sufficient to reconstruct exactly the original scene using back-projection techniques. The goal of these techniques is to find all the disparity correspondences between the different views. Once this disparity matching problem is solved, the exact depth of any object can be retrieved and the original scene can be reconstructed exactly. We consider here three scene scenarios leading to three different sampling techniques:

**A:** We first consider the case where the $L$ planar objects are separated and visible from all the $N$ cameras without occlusion, and keep the same "left to right" ordering (that is, the $k^{th}$ piece on the $i^{th}$ view corresponds to the $k^{th}$ piece on the $j^{th}$ view, for any $k \in \{1; L\}$ and $i, j \in \{1; N\}$). With this hypothesis, only two of the $N$ views are necessary in order to reconstruct the original scene (the correspondence problem between the two views is straightforward in this case).

**B:** We then consider the case where the $L$ objects are separated and visible from all the $N$ cameras without occlusion, but where the "left to right" ordering is not guaranteed anymore. In order to solve the disparity matching problem at the decoder, we need to have at least $L + 1$ views of the scene. This is a sufficient condition to guarantee a correct matching of the $L$ objects in order to retrieve their real positions in the scene. It is easy to observe that if all the objects are similar (say black dots), the matching of less than $L + 1$ views would lead to an ambiguous reconstruction where several solutions are possible (back-projection problem) [10]. In our case, the projections (or views) are made of polynomial pieces and can be correctly matched using at least $L + 1$ projections. Practically, we can back-project the $L + 1$ left extremities of the pieces and retrieve the $L$ real locations of these extremities. The

same procedure is then repeated with the $L+1$ right extremities. Notice that this approach only relies on the discontinuity locations and not on the intensity of the pieces. This general sampling result is therefore sufficient in any case (even if all the pieces have the same intensity), but is not always necessary (two views are in theory sufficient if all the pieces have different intensities).

**C:** Finally, we also assume the following hypothesis regarding possible occlusions:

- each object can be occluded in at most $O_{max}$ views. (For the sake of simplicity, we only consider full occlusions here. Any object can thus only either be fully visible or fully occluded in any given view.)

The number of sensors needed in order to guarantee an exact reconstruction of the scene is given by $N_{min} \geq L + O_{max} + 1$. (Note that in practice, we can expect $O_{max}$ to scale linearly with the total number of sensors $N$.)

For these three scenarios, the minimum number of cameras corresponds to the critical sampling. In the next section, we show how each sensor can quantize its parameters and use a distributed compression approach to maintain a constant global rate-distortion behaviour at the decoder, independent on the number of sensors involved in the transmission of information to the receiver.

## 4.3 An Exact Bit-Conservation Principle

The distributed compression algorithm applied at each sensor can be summarized as follows: First, the original projection is reconstructed from the observed sampled version using an FRI reconstruction method. The original view parameters retrieved (which are the $2L$ discontinuity locations and the $Q+1$ coefficients of the $L$ polynomials of maximum degree $Q$) are then scalar quantized according to some target

distortion value for the reconstructed view. Finally, each quantized parameter is S-W encoded according to the known correlation structure between the different views. Our distributed source coding approach proposed in the previous chapter can easily be applied in this context of piecewise polynomial signals. Each signal is represented by its set of discontinuities (corresponding to the objects locations) and its set of polynomials (corresponding to the objects intensities). These polynomials are common to all the views (in the case of no occlusion) and the discontinuities can be S-W encoded as proposed in Chapter 3.

In this section, we first describe the rate-distortion behaviour of our view model when the view parameters are encoded independently. Then, we introduce in more details the correlation between the different views and show that our distributed compression approach guarantees a bit-conservation principle. Notice that for the sake of simplicity we continue to focus on the 1-D scenario only (correlated horizontal scanlines).

### 4.3.1 R-D behaviour of independent encoding

A 1-D view is modelled by a piecewise polynomial function defined on $[0;T]$ with $L$ independent pieces of maximum degree $Q$, bounded in amplitude in $[0,A]$, and $2L$ discontinuities. Assume that such a function is quantized using $R_t$ and $R_p$ bits to represent each discontinuity and polynomial piece respectively (the parameters are quantized using uniform scalar quantizers). It is possible to show that the distortion (MSE) of its reconstruction can be bounded with the following expression, using the results presented in [55]:

$$D(R_p,R_t) \leq \frac{1}{2}A^2LT((Q+1)^2\ \ 2^{-\frac{2}{Q+1}R_p}+2^{-R_t}). \tag{4.1}$$

For a total number of $R = L(2R_t + R_p)$ bits, the optimal bit allocation is given by: $R_p = \frac{Q+1}{Q+5}\frac{R}{L} + G$ and $R_t = \frac{2}{Q+5}\frac{R}{L} - \frac{1}{2}G$, where $G = 2\frac{Q+1}{Q+5}(\log(Q+1)+2)$. This allocation leads to the following optimal rate-distortion behaviour:

$$D(R) \leq \underbrace{\frac{1}{2}A^2LT((Q+1)^2\ 2^{-\frac{2}{Q+1}G} + 2^{\frac{1}{2}G})}_{c_0}\, 2^{\frac{-2}{L(Q+5)}R}. \tag{4.2}$$

A more detailed derivation of this rate-distortion bound is available in Appendix A.1.

### 4.3.2 R-D behaviour using our distributed coding approach

Assume $f_1(t)$ and $f_2(t)$ are two 1-D piecewise polynomial views obtained from two different cameras that are at a distance $\alpha$ apart. The two signals are defined for $t \in [0;T]$ and are bounded in amplitude in $[0;A]$. If there is no occlusion, the two views are exactly represented by $L$ polynomials of maximum degree $Q$, and $2L$ discontinuities (the signals are equal to zero between the pieces). The shift of a discontinuity from one view to the other (its disparity) is given by the epipolar geometry and can be defined as: $\Delta_i = \frac{\alpha f}{z_i}$, where $z_i$ is the depth of the $i^{th}$ discontinuity (the depth of the object at its extremity). The range of possible disparities for a scene is therefore given by: $\Delta \in [0; \frac{\alpha f}{z_{min}}]$ (see Figures 3.3 and 4.1).

We assume Lambertian surfaces for all the planar objects that make up the scene (i.e., the intensity of any point on the surface remains the same when observed from different viewing positions). A polynomial piece corresponding to a tilted planar object can be linearly contracted or dilated in the different views. However, its representation using Legendre polynomials is the same for any view, because of the support normalization on $[-1;1]$ that we use with this basis.

The correlation between the two views is therefore such that, knowing all the parameters of the first view, the only missing information necessary to reconstruct

perfectly the parameters of the second view is the set of disparities $\{\Delta_i\}_{i=1}^{2L}$. The information about these disparities can be encoded in a distributed manner using an approach similar to the one presented in Chapter 3.

Each sensor has to quantize and transmit its parameters to the central receiver. In the case where there is no occlusion, this information corresponds exactly to $2L$ discontinuity locations and $L$ polynomials. Let $R_{t_i}$ and $R_{p_i}$ be the number of bits used to quantize each discontinuity and each polynomial of the $i^{th}$ view respectively. The total number of bits used to encode this view is therefore given by $R_i = L(2R_{t_i} + R_{p_i})$ and is associated to a distortion $D_i$.

Assume just for one instant that the $N$ views that have to be transmitted to the receiver are independent. In this case, for an average distortion $D_i$ over all the reconstructed views at the receiver, the total amount of data to be transmitted would be of the order $NR_i = NL(2R_{t_i} + R_{p_i}) \rightarrow \sum_{i=1}^{N} L(2R_{t_i} + R_{p_i})$.

We can now show how the correlation model can be exploited to perform distributed compression for the three scene scenarios introduced in Section 4.2:

**A:** In this first scenario, we know that the knowledge of the discontinuity locations and the polynomial coefficients from only two cameras is sufficient to retrieve the original scene parameters. The correlation model tells us that the location of the discontinuities on two different views are such that the range of possible disparities is given by $\Delta_i \in [0; \frac{\alpha f}{z_{min}}]$, where $\alpha$ is the distance between the two considered cameras. The polynomial intensities of the different pieces are also known to be the same for the two views (Lambertian surface assumption). This strong correlation structure can therefore be exploited by means of a distributed coding approach.

Assume that a total of $R$ bits is used to encode the first signal such that each polynomial piece and each discontinuity is represented using $R_p$ and $R_t$ bits respectively ($R = L(R_p+2R_t)$). On receiving this information, the decoder can reconstruct

the first view with a distortion $D$. Assume now that the second encoder also uses $R$ bits to encode the other view at a similar distortion. Knowing that the encoded version of the first view is already available at the decoder, the second encoder does not need to transmit all its encoded data.[1] First of all, since the polynomial coefficients are similar for the two views, they do not need to be re-transmitted from the second encoder. Secondly, since the discontinuity locations are correlated, only partial information from each discontinuity need to be transmitted to the decoder using a distributed coding approach similar to the one proposed in Chapter 3. The total number of bits required from the second encoder therefore corresponds to $2LR_{t_{SW}} = 2L(R_t - \gamma_s)$ bits, where $\gamma_s = \lfloor \log_2(\frac{T}{\Delta_{max}}) \rfloor$ corresponds to the number of most significant bits of each discontinuity location that does not need to be transmitted from the second view (Figure 4.2 illustrates this asymmetric distributed coding strategy).

Thus, the total number of bits necessary to transmit the two views is given by $R_{tot} = L(R_p + 2R_t + 2R_{t_{SW}})$. Using this encoded data, the decoder can reconstruct the two views or any other view with a distortion of the order of $D$. Notice that the allocation of the bit-rates amongst the two encoders is very flexible as illustrated in Figure 4.3 (symmetric case).

Assume now that we want to transmit information of the scene from more than two encoders in order to balance the communication task amongst a larger set of available cameras. The total information necessary at the decoder to reconstruct all the different views with a distortion $D$ can be divided and partially obtained from any set of cameras using again the approach proposed in Chapter 3. We know that the information about the polynomials can be arbitrarily obtained from any camera and that as long as the two most distant cameras transmit their $R_{t_{SW}}$ least

[1] Notice that we assume a high bit-rate regime where the quantization errors are small compared to the amplitudes of the disparities and the polynomials, such that the quantized view parameters still satisfy (or nearly satisfy) the correlation model.

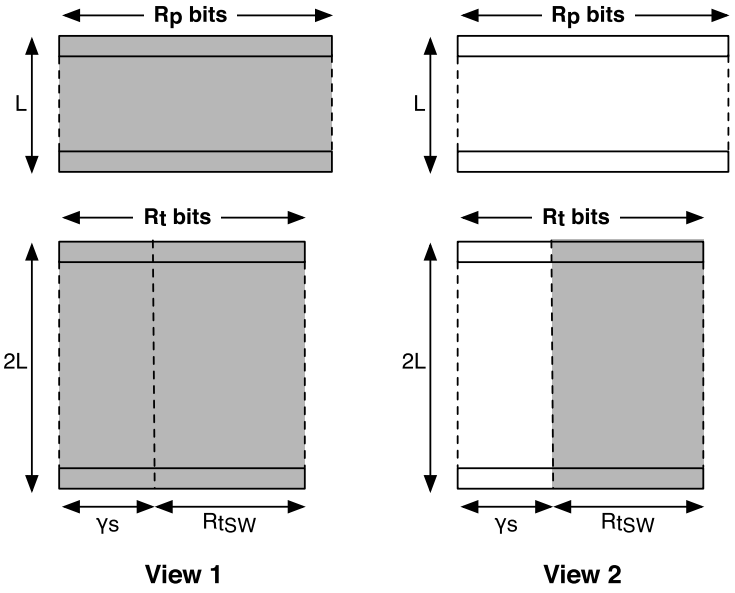


**Figure 4.2: Distributed asymmetric encoding of two views. Each view is first encoded using $R = L(R_p + 2R_t)$ bits. Encoder 1 transmits its full representation using $R$ bits. Encoder 2 only transmits the $R_{t_{SW}}$ least significant bits of its $2L$ discontinuity locations (gray area).**

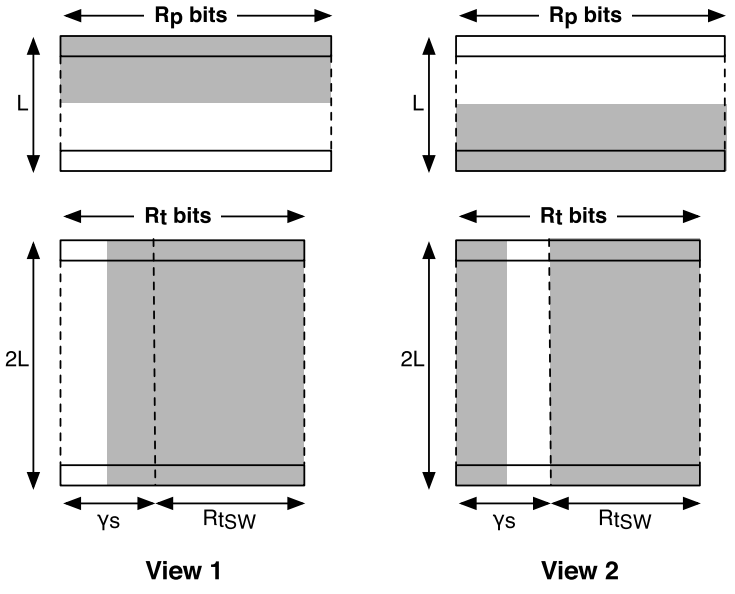


**Figure 4.3: Distributed symmetric encoding of two views. Complementary subsets of the $LR_p$ bits representing the polynomials and the $2L\gamma_s$ bits representing the most significant bits of the discontinuities are transmitted from the encoders, along with all the $2LR_{t_{SW}}$ least significant bits of their discontinuities.**

significant bits for each of their $2L$ discontinuities, the rest of the $R_{tot} - 2LR_{t_{SW}}$ bits are common and can be obtained from any subset of the $N$ encoders (Figure 4.4 illustrates a distributed coding strategy using five cameras). In other words, once the two extreme sensors have transmitted their $R_{t_{SW}}$ least significant bits of each of their $2L$ discontinuities, the remaining $L(R_p + 2\gamma_s)$ bits to be transmitted (which is the information that is common to all sensors) can be obtained from any subset of the $N$ cameras.[2] The number of sensors used for the transmission has therefore

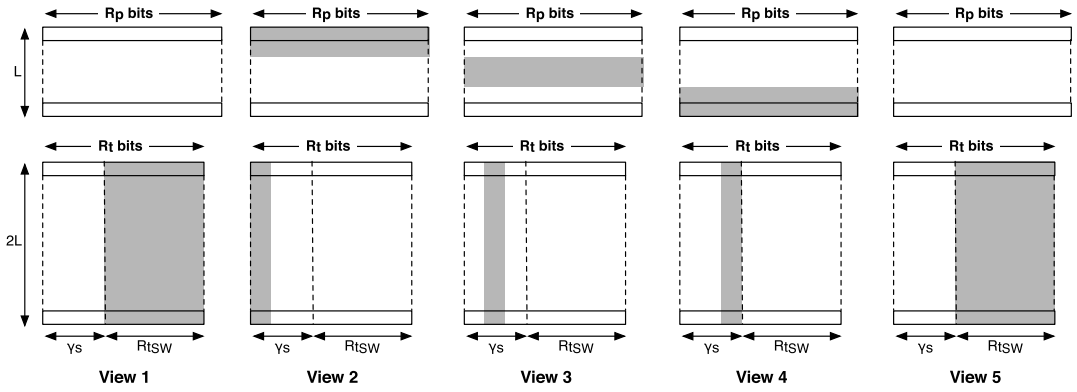


**Figure 4.4: Distributed encoding of five views. Encoders 1 and 5 transmit the $R_{t_{SW}}$ least significant bits of each of their $2L$ discontinuities. Complementary subsets of the $LRp$ bits representing the polynomials and the missing $2L\gamma_s$ bits from the discontinuities are obtained from encoders 2, 3 and 4.**

no influence on the reconstruction fidelity at the decoder. Using an optimal bit allocation and transmitting the information using $N$ sensors, the distortion of any reconstructed view given the total bit budget $R_{tot}$ can be shown to behave as (see Appendix A.2):

$$D_A(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2(2\gamma_s+G)}{Q+9}}}_{c_1} 2^{\frac{-2}{L(Q+9)}R_{tot}}. \tag{4.3}$$

Notice that an independent encoding of the N views would lead to the following behaviour:

$$D(R_{tot}) \leq c_0 2^{\frac{-2}{NL(Q+5)}R_{tot}}. \tag{4.4}$$

[2]Notice that in order to perform a symmetric encoding over the $N$ cameras, we need to guarantee a high bit-rate regime where $\frac{R_{tot}}{N} \geq 2LR_{t_{SW}}$.

We can observe that Eq.(4.3) does not depend on the number of sensors $N$.

**B:** The distributed compression strategy in this case consists in sending the discontinuity locations from $L+1$ views and each polynomial piece from only one encoder. The total bit-rate necessary is therefore given by: $R_{tot} = L((L+1)2R_t + R_p)$. If we now want to transmit information from more than this minimum number of sensors $N_{min} = L+1$, we can do it in a flexible manner: For each new sensor introduced in the system, we can ask it to take the responsibility of transmitting partial information about the polynomial pieces (therefore reducing the communication task of some other sensors) or to replace one of its two neighbours to transmit some subset of the most significant bits of its discontinuity locations. The distortion of any reconstructed view given the total bit budget $R_{tot}$ can be shown to behave as (see Appendix A.3):

$$D_B(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2LG}{4L+Q+5}}}_{c_2} 2^{\frac{-2}{L(4L+Q+5)}R_{tot}}. \tag{4.5}$$

Again, the distortion at the receiver only depends on the total number of bits transmitted $R_{tot}$ and not on the number of sensors used.

**C:** The distributed compression strategy for this scenario can be summarized as follows:

- Transmit the position of the discontinuities from $L + O_{max} + 1$ views.
- Transmit the polynomial pieces from $O_{max} + 1$ views.

The total bit-rate necessary is therefore given by: $R_{tot} = L((L + O_{max} + 1)2R_t + (O_{max} + 1)R_p)$. The flexible repartition of the transmission bit-rate used in the previous scenario still holds here when more sensors are used for the transmission.

The distortion of any reconstructed view given the total bit budget $R_{tot}$ can be shown to behave as (see Appendix A.4):

$$D_C(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2O_{max}G}{4L+(O_{max}+1)(Q+5)}}}_{c_3} 2^{\frac{-2}{4L^2+L(Q+5)(O_{max}+1)}R_{tot}}. \tag{4.6}$$

Table 4.1 summarizes the different R-D bounds derived in this chapter (see Appendix A for more details).

**Table 4.1: Summary of the different R-D bounds.**

| Encoding mode | total number of bits $R_{tot}$ (for the encoding of $N$ views) | average R-D upper bounds |
|---|---|---|
| Indep. Encoding | $R_{tot} = NL(2R_t + R_p)$ | $D(R_{tot}) \leq c_0 2^{\frac{-2}{NL(Q+5)}R_{tot}}$ |
| DSC - Scenario A | $R_{tot} = L(R_p + 4R_t - 2\gamma_s)$ | $D_A(R_{tot}) \leq c_1 2^{\frac{-2}{L(Q+9)}R_{tot}}$ |
| DSC - Scenario B | $R_{tot} = L((L+1)2R_t + R_p)$ | $D_B(R_{tot}) \leq c_2 2^{\frac{-2}{L(4L+Q+5)}R_{tot}}$ |
| DSC - Scenario C | $2L^2R_t + L(O_{max}+1)(R_p + 2R_t)$ | $D_C(R_{tot}) \leq c_3 2^{\frac{-2}{4L^2+L(Q+5)(O_{max}+1)}R_{tot}}$ |

## 4.4 Simulation Results

We propose a simple simulation where five cameras are observing a synthetic scene made of three tilted polygons with linear intensities. The cameras are placed on a horizontal line and observe blurred and undersampled views ($32 \times 32$ pixels) of the original scene as illustrated in Figure 4.5. For each view, knowing that the original scene belongs to a certain class of signals with finite rate of innovation, the sampling results presented in Section 4.2 can be used to retrieve the 33 original parameters of the view (i.e. twelve vertices having two parameters each, and three 2-D linear functions having three parameters each). Using these retrieved parameters, high resolution version of the different views can be reconstructed at each encoder (see Figure 4.6).

The original views are represented with the following precision: 22 bits for each vertex (view of $2048 \times 2048$ pixels) and 9 bits for each polynomial coefficient. One

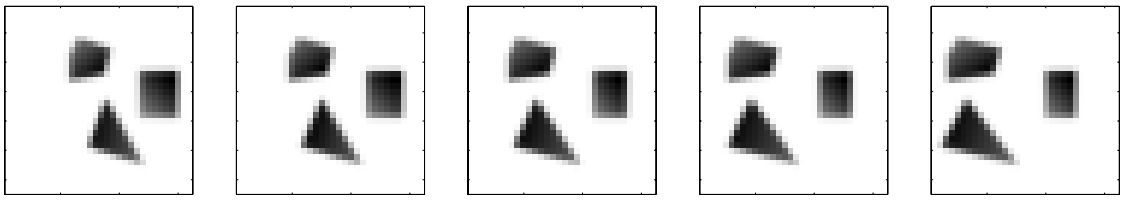

**Figure 4.5: Observations at the sensors. Each sensor observes a blurred and undersampled version of the perspective projection of the scene.**

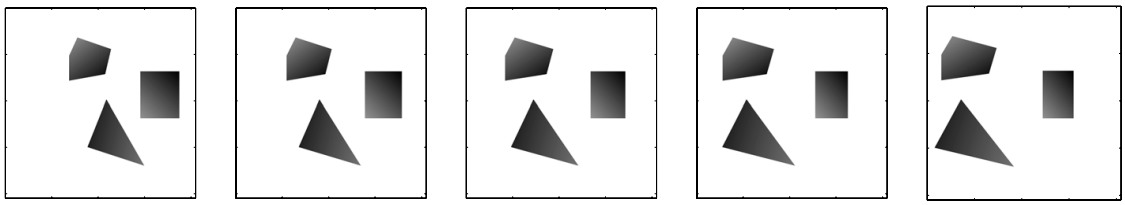

**Figure 4.6: High resolution version of the 5 views reconstructed at each encoder from the samples shown in Figure 4.5 using the FRI method of [68].**

view is therefore exactly represented using $12\times22+9\times9 = 345$ bits. The parameters $\alpha$, $f$ and $z_{min}$ are such that $R_{t_{SW}} = 10$ bits (the disparities between the first and the fifth views can actually be larger than a quarter of the image width). As we have shown in Section 4.3.2, a total rate of $R_{tot} = 345 + 12 \times 10 = 465$ bits is therefore sufficient to reconstruct all the high resolution views at the receiver.

If the bit budget $R_{tot}$ is smaller than 465, scalar quantization of the parameters has to be done prior to applying distributed compression. Table 4.2 highlights our bit-conservation principle for different bit budgets $R_{tot}$. In particular, it shows that our approach suffers no rate loss when we increase the number of sensors used for transmission.

**Table 4.2: An exact bit-conservation principle: simulation results with the 5 views presented in Figure 4.5.**

| $\mathbf{R_1}$ (bits) | $\mathbf{R_2}$ (bits) | $\mathbf{R_3}$ (bits) | $\mathbf{R_4}$ (bits) | $\mathbf{R_5}$ (bits) | $\mathbf{R_{tot}}$ (bits) | **PSNR** (dB) |
|---|---|---|---|---|---|---|
| 345 | - | - | - | 120 | 465 | $\infty$ |
| 276 | - | - | - | 108 | 384 | 38.8 |
| 128 | - | 128 | - | 128 | 384 | 38.8 |
| 108 | 84 | - | 84 | 108 | 384 | 38.8 |
| 84 | 72 | 72 | 72 | 84 | 384 | 38.8 |
| 171 | - | - | - | 60 | 231 | 22.8 |
| 48 | 48 | 39 | 48 | 48 | 231 | 22.8 |

## 4.5 Conclusions

In this chapter, we have addressed the problem of distributed sampling and compression in camera sensor networks for scenes with finite rate of innovation. In particular, we have shown that if a piecewise polynomial model is used to represent the scene, the critical sampling (i.e., the minimum number of sensors needed) can be precisely derived. Then, we have also shown that when the number of sensors increases, a distributed compression approach can be used to maintain a constant global rate usage (independent on the number of sensors $N$), for any given reconstruction quality. This result therefore leads to an exact bit-conservation principle.

# Chapter 5

# Geometry-Driven Distributed Compression of Real Multi-view Images

## 5.1 Introduction

In this chapter, we show how a particular image coder based on tree structured algorithms [69] can be modified to take advantage of our distributed coding approach proposed in Chapter 3. Notice that several other distributed compression approaches for real multi-view images have been proposed (see Section 2.3.3 for a more detailed description of these methods). Our approach differs from the others in that it tries to estimate the correlation structure in the visual information using some geometrical constraints, and does not use conventional channel codes. The main properties of our scheme is that a) it allows for an arbitrary partition of the rate between the encoders and b) it satisfies a bit conservation principle. Namely, the reconstruction fidelity at the decoder does not depend on the number of sensors involved, but only on the total bit-budget $R$ (as presented in Chapter 4).

## 5.2 Distributed Compression using Tree Structured Approaches

This section introduces the model that we use to represent the different views acquired from the linear multi-camera set-up that we consider in our work (see Figure 3.3). Then, we present a practical algorithm that can be used to approximate real multi-view images with the proposed model. Finally, we show how distributed compression can be performed on the approximated views in the 1-D (i.e., compression of scan-lines of different views) and in the 2-D case.

### 5.2.1 Multi-view model using piecewise polynomial representations

Since the correlation model used by our distributed coding approach is related to the object's positions on the different views (see Chapter 3), we need to develop a coding algorithm that can efficiently represent these positions. Our approach consists in representing the different views using a piecewise polynomial model. The main advantage of such a representation is that it is well adapted to represent real images and that it is able to precisely catch the discontinuities between objects. Two different views can therefore be modeled using a piecewise polynomial signal where each discontinuity is shifted according to the correlation model: $\Delta_i \in \{\Delta_{min}, \Delta_{max}\}$. Moreover, if we assume that the scene is composed of lambertian planar surfaces only and that no occlusion occurs in the different views, then we can claim that the polynomial pieces are similar in the different views.[1] Notice that this piecewise polynomial model was already used in Chapter 4, where we assumed that the different

[1]With non-lambertian surfaces, or with the presence of occlusions, the polynomial pieces can differ in the different views. Our simple correlation model should therefore be modified in this case. For the sake of simplicity, we will however only consider this simple model to present our coding approach.

views were signals with FRI that could exactly be represented by this model. In this chapter, we will show how real multi-view images can be approximated and compressed with this model by means of a tree-based decomposition approach. Then, our distributed source coding method will be used to compress these aproximations in a distributed manner.

### 5.2.2 The prune-Join decomposition algorithm

In [69], Shukla et al. presented new coding algorithms based on tree structured segmentation that achieve the correct asymptotic rate-distortion (R-D) behaviour for piecewise polynomial signals. Their method is based on a prune and join scheme that can be used for 1D (using binary trees) or for 2D (using quadtrees) signals in a similar way.

The aim of this coding approach (in 1-D) is to approximate a given signal using a piecewise polynomial representation. The first step of the algorithm consists in segmenting the input signal using a binary tree decomposition. The signal is first split in two pieces of the same length, then each piece is split again in a recursive manner until the whole signal is decomposed into $2^{J_{max}}$ pieces of length $T2^{-J_{max}}$ where $T$ is the support of the original signal. The binary tree representing this full decomposition therefore consists of $2^{J_{max}+1}-1$ nodes, each corresponding to a certain region of the signal with a certain length. The second step of the algorithm consists in approximating each node of the binary tree (i.e. a given region of the signal) with a polynomial function of maximum degree $Q$. The best polynomial approximation is obtained using a pseudo-inverse approach and is therefore optimal in the mean squared error sense. The third step of the algorithm consists in generating R-D curves for each node of the binary tree by quantizing the polynomial coefficients of the approximations using a number of bits in the range $[0; R_{max}]$. Then, assuming that a global bit budget of $R$ bits is available to encode the whole signal, optimal bit

allocation between the different nodes of the binary tree must be performed. The optimal bit allocation is obtained by choosing a fixed operating slope $\lambda$ to select the current rate on all R-D curves and performing the pruning of children nodes when the following Lagrangian cost based criterion holds: $(D_{c_1} + D_{c_2}) + \lambda(R_{c_1} + R_{c_2}) \geq (D_p + \lambda R_p)$. The indices $c_1$, $c_2$ and $p$ correspond to the two children and the parent node respectively. The resulting pruned tree gives the sequence of leaves that represent the optimal bit allocation according to the current operating slope $\lambda$. In order to further improve the encoding performances, neighbouring leaves that do not have the same parent node are joined and coded together when the following Lagrangian cost based criterion holds: $(D_{n_1} + \lambda R_{n_1}) + (D_{n_2} + \lambda R_{n_2}) \geq (D_{n_{Joint}} + \lambda R_{n_{Joint}})$. The resulting encoded version of the original signal is therefore represented using the pruned tree, the joining information and the polynomial coefficients of each approximated region (i.e. group of joined leaves). The pruned tree is represented using a number of bits corresponding to its number of nodes (scanning the tree with a top-down, left-to-right approach, a 0 means that the node has children, while a 1 indicates that the node is actually a leaf). The joining information requires a number of bits corresponding to the number of leaves in the pruned tree (each leaf needs to indicate if it is joined to the next one or not). If the total number of bits used to encode the signal does not correspond to the global bit budget $R$, the operating slope $\lambda$ must be updated and the pruning and joining procedure must be repeated until it reaches the global targeted rate or distortion.

We give here a sketch of this compression algorithm for 1D signals (Algorithm 5.1) and encourage the reader to refer to the original work [69] for more details.

**Algorithm 5.1** Prune-Join binary tree coding algorithm

1: Segmentation of the signal using a binary tree decomposition up to a tree depth $J_{max}$.
2: Approximation of each node of the tree by a polynomial $p(t)$ of degree $\leq Q$.
3: Rate-Distortion curves generation for each node of the tree (scalar quantization of the Legendre polynomial coefficients).
4: Optimal pruning of the tree for the given operating slope $-\lambda$ according to the following Lagrangian cost based criterion: Prune the two children of a node if $(D_{c_1} + D_{c_2}) + \lambda(R_{c_1} + R_{c_2}) \geq (D_p + \lambda R_p)$.
5: Joint coding of similar neighbouring leaves according to the following Lagrangian cost based criterion: Join the two neighbours if $(D_{n_1} + \lambda R_{n_1}) + (D_{n_2} + \lambda R_{n_2}) \geq (D_{n_{Joint}} + \lambda R_{n_{Joint}})$.
6: Search for the desired R-D operating slope (update $\lambda$ and go back to point 4).

### 5.2.3 Our distributed coding strategy for 1-D piecewise polynomial functions

Let $f_1(t)$ be a piecewise polynomial signal defined over $[0; T]$ consisting of $L + 1$ polynomial pieces of maximum degree $Q$ each, and bounded in amplitude in $[0; A]$. Let $\{t_{1,i}\}_{i=1}^{S}$ represent the $S$ distinct discontinuity locations of $f_1(t)$. We define $f_2(t)$ as another piecewise polynomial function over $[0; T]$ having the same polynomial pieces than $f_1(t)$, but the discontinuity locations $\{t_{2,i}\}_{i=1}^{S}$ are such that: $\Delta_{min} \leq t_{2,i} - t_{1,i} \leq \Delta_{max}, \forall i \in \{1, \ldots, S\}$. The relationship between $f_1(t)$ and $f_2(t)$ is therefore given by the range of possible disparities $[\Delta_{min}; \Delta_{max}]$ which corresponds to the plenoptic constraints we consider in our camera sensor network scenario.

Assume that these two signals are independently encoded using Algorithm 5.1 for a given distortion target. The total information necessary to describe each of them can be divided in three parts: $R_{Tree}$ is the number of bits necessary to code the pruned tree and is equal to the number of nodes in the tree. $R_{LeafJointCoding}$ is the number of bits necessary to code the joining information and is equal to the number of leaves in the tree. Finally, $R_{Leaves}$ is the total number of bits necessary to code the set of polynomial approximations.

Figure 5.1 presents the prune-join tree decompositions of two piecewise constant signals having the same set of amplitudes and having their sets of discontinuities satisfying our plenoptic constraints. Because of these constraints, we can observe that the structure of the two pruned binary trees has some similarities. Our distributed compression algorithm uses these similarities in order to transmit only the necessary information to allow for a complete reconstruction at the decoder. It can be described as follows (asymmetric encoding):

- Send the full description of signal 1 from encoder 1 using $R_1$ bits. ($R_1 = R_{Tree_1} + R_{LeafJointCoding_1} + R_{Leaves_1}$)
- Send only the subtrees of signal 2 having a root node at level $J_\Delta$ along with the joining information from encoder 2 using $R_2$ bits. Here, $J_\Delta = \lceil \log_2(\frac{T}{\Delta_{max}-\Delta_{min}+1}) \rceil$. Therefore, $R_2 = (R_{Tree_2} - R_\Delta) + R_{LeafJointCoding_2}$ where $R_\Delta$ corresponds to the number of nodes in the pruned tree with a depth smaller than $J_\Delta$.

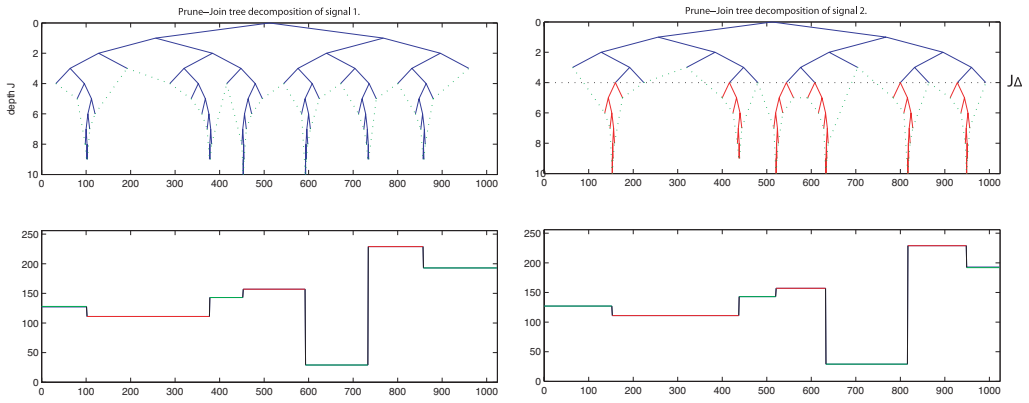


**Figure 5.1: Prune-Join binary tree decomposition of two piecewise constant signals satisfying our correlation model.**

At the decoder, the original position of the subtrees received from encoder 2 can be recovered using the plenoptic constraints (i.e. $\Delta \in [\frac{\alpha f}{z_{max}}; \frac{\alpha f}{z_{min}}]$) and the side information provided by encoder 1. The full tree can then be recovered and the

second signal can thus be reconstructed using the set of amplitudes received from encoder 1.

### Arbitrary bit-rate allocation

The construction proposed in the previous subsection is asymmetric since encoder 1 has to transmit its whole information, whereas encoder 2 only transmits missing information from its pruned tree structure and its joining information. In order to allow for an arbitrary bit-rate allocation between the two encoders, we propose the following strategy:

- Send all the subtrees having a root node at level $J_\Delta$ from both encoders, along with their joining information.
- Send complementary parts of the two upper trees (depth $< J_\Delta$).
- Send complementary subsets of the polynomial approximations.

The complementary subsets of the polynomial approximations can be chosen in a fully flexible manner, but has to be static and known originally by the encoders, as they cannot communicate between themselves. The subsets can typically consist of sets of complementary bit-planes, complementary spatial regions of the signal, or a combination of both. For example, it can be decided that encoder 1 only transmits the most significant bits of its polynomial coefficients where encoder 2 transmits the complementary least significant bits. This coding approach presents the same global rate-distortion behaviour than the asymmetric approach, but allows for an arbitrary bit-rate allocation (see Figure 5.2 and Table 5.1 for simulation results).

### An exact bit conservation principle

The distributed compression approach presented before can be extended to more than two cameras. We consider the multi-camera scenario presented in Figure 3.3 with $N$ cameras. If we consider that there is no occlusion in these different views, the knowledge of the discontinuity locations from only two cameras is sufficient to reconstruct any view in between (see Scenario A in Chapter 4). Assume that we have a total bit budget of $R_{tot}$ bits to allocate between the different encoders and that only the two most distant cameras are transmitting information to the decoder. In this case, we know that each encoder should first compute a representation of its input signal using $R = R_{Tree} + R_{LeafJointCoding} + R_{Leaves}$ bits such that $(R_{Tree} - \frac{1}{2}R_{\Delta}) + R_{LeafJointCoding} + \frac{1}{2}R_{Leaves} = \frac{1}{2}R_{tot}$ (symmetric encoding). At the decoder, the two views can be reconstructed and any new view in between can be interpolated with a certain fidelity.

Assume now that we want to transmit some information from other cameras as well. We can show that, as long as the subtrees (depth $\geq J_{\Delta}$) are transmitted from the two extreme cameras, the remaining information is common to all cameras and can therefore be obtained from any set of cameras [22]. Here, the subtrees represent the "syndrome" of the discontinuity locations. They correspond therefore to the innovation that cannot be obtained from the discontinuities location of any other view. Moreover, because of the pixel precision of these locations, using the two most distant cameras to retrieve them will lead to the best interpolation result at the other camera locations. Then, since the polynomial pieces are similar for each view, they can be transmitted from any camera, all this without impairing reconstruction quality. This result leads to a practical algorithm reaching the exact bit conservation principle of Chapter 4 (see Figure 5.3 and Table 5.2 for simulation results).

### 5.2.4 Simulation results on 1-D Piecewise polynomial signals

We have applied our distributed compression approach to different sets of 1-D piecewise polynomial signals in order to highlight the arbitrary bit-rate allocation and the bit conservation principle presented in Section 5.2.3. In Table 5.1, we show that an independent encoding of the two signals presented in Figure 5.2 requires a total bit-rate of .304 bpp to achieve a distortion target (SNR) of 26 dB. Using our distributed compression approach, we can see that a quarter of the total bit-rate can be saved for an identical reconstruction fidelity. Moreover, this result remains constant for any choice of bit-rate allocation (SW asym. or SW sym.). Notice that a similar reduction of bit-rate with an independent encoding strategy would see the total SNR to drop by about 10 dB.

**Table 5.1: Arbitrary bit-rate allocation: simulation results with two 1-D views.**

| **Coding Strategy** | $\mathbf{R_1}$ (bpp) | $\mathbf{R_2}$ (bpp) | $\mathbf{R_{tot}}$ (bpp) | $\mathbf{D_{tot}}$ **(SNR)** (dB) |
|---|---|---|---|---|
| Independent | .307 | .301 | .304 | 26.01 |
| SW asym. 1 | .307 | .148 | .228 | 26.01 |
| SW asym. 2 | .154 | .301 | .228 | 26.01 |
| SW sym. | .227 | .227 | .227 | 26.01 |

In Table 5.2, we highlight our bit conservation principle, by applying our compression approach to the three views shown in Figure 5.3. We know that the two extreme views contain enough information to allow the reconstruction of any view in between with a comparable fidelity. Applying our S-W approach to these two extreme views, a total of 235 bits is necessary to achieve a distortion (SNR) of about 26 dB for the three reconstructed views. A similar global rate-distortion behaviour holds when part of the information is transmitted from the central view.[2] This result highlights our bit conservation principle. In other words, if we assume that the

[2]The small variations in the SNR values are only due to different quantization errors.

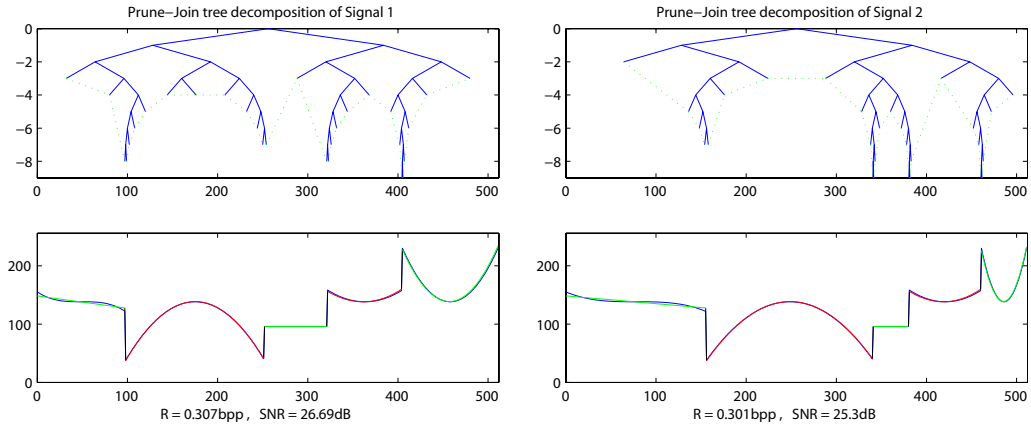


**Figure 5.2: Join-Prune tree decomposition of two piecewise polynomial signals with shifted discontinuities for a given distortion target (26 dB).**

sensors transmit their compressed data to the central decoder using a multi-access channel, the fidelity of the reconstructed views only depends on the global capacity of this channel and not on the number of sensors used.

**Table 5.2: An exact bit conservation principle: simulation results with three 1-D signals.**

| **Coding Strategy** | $\mathbf{R_1}$ (bits) | $\mathbf{R_2}$ (bits) | $\mathbf{R_3}$ (bits) | $\mathbf{R_{tot}}$ (bits) | $\mathbf{D_{tot}}$ **(SNR)** (dB) |
|---|---|---|---|---|---|
| Independent | 157 | 124 | 157 | 438 | 26.13 |
| SW - 2 views | 117 | 0 | 118 | 235 | 25.68 |
| SW - 3 views | 82 | 71 | 82 | 235 | 26.13 |

### 5.2.5 Simulation results on scan lines of stereo images

In order to justify the correlation model we used in our distributed compression scheme, we applied it to a set of scan lines of real multi-view images. We present a simulation on scan lines of a pair of stereo images (Figure 5.4) using a piecewise linear model and a symmetric encoding strategy. The reconstructed signals present a good level of accuracy for the discontinuity locations. However, since the assumption of lambertian surfaces does not hold for this scene, the polynomial pieces can sometime

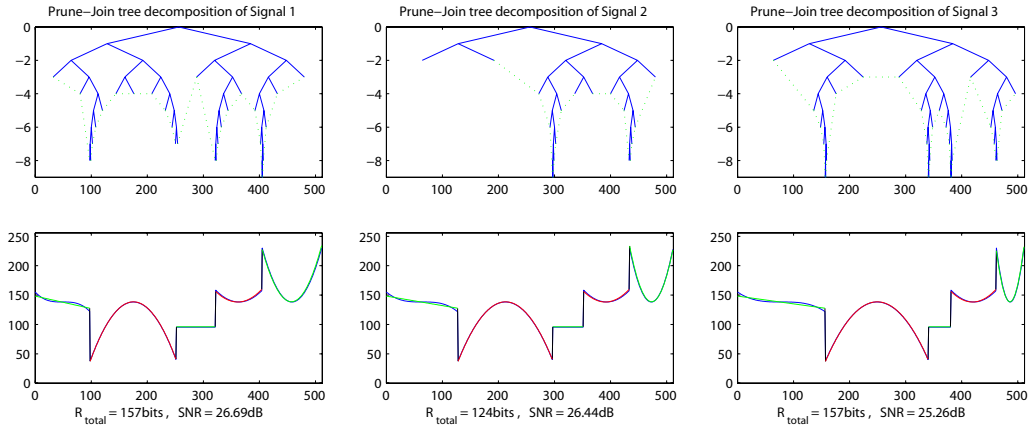


**Figure 5.3: Join-Prune tree decomposition of three piecewise polynomial signals for a given distortion target (26 dB).**

be slightly different from one view to the other. A solution to this problem would be to consider a more sophisticated correlation model for the polynomials.

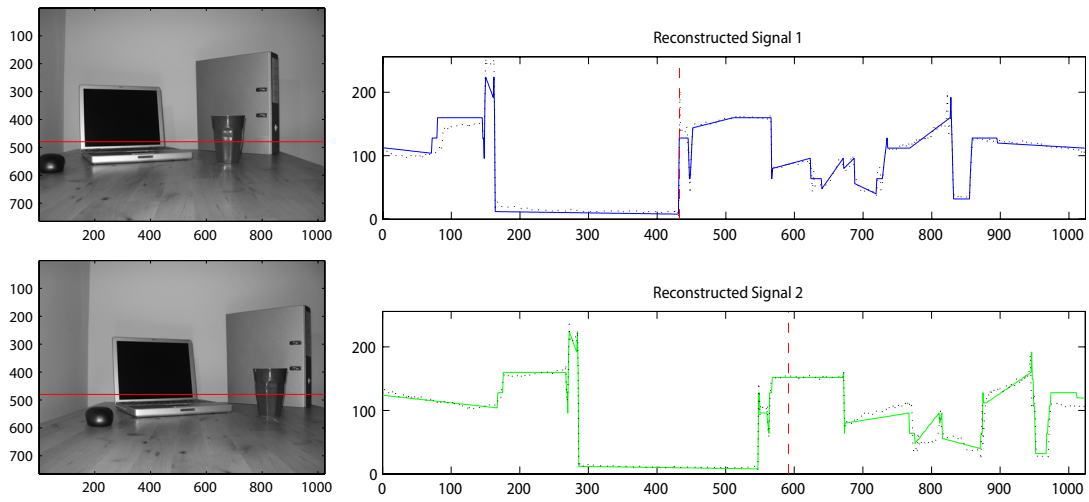


**Figure 5.4: (left) Stereo images of a real scene where the objects are located between a minimum and a maximum distance from the cameras. (right) Reconstructed scan lines using a piecewise linear model for the binary tree decomposition and a symmetric distributed compression.**

## 5.3 Extension to 2D using Quadtree Decompositions

In this section, we extend our results by showing how the prune-join binary tree decomposition used in the 1D case has an intuitive extension in 2D. In this case, the binary tree is replaced with a quadtree and the polynomial model is replaced with a 2D geometrical model. Our implementation of the quadtree encoder, which is directly inspired from the prune-join quadtree decomposition algorithm proposed in [69], is first presented and we then show how our distributed coding scheme can be adapted in this context. Our simulation results show that our approach still outperforms the rate-distortion behaviour of independent encoding with real multi-view data even when the correlation model is not fully respected.

### 5.3.1 The prune-join quadtree compression algorithm and the geometrical model used

We give here a sketch of our implementation (Algorithm 5.2) of the quadtree compression approach proposed in [69]. Figure 5.5(b) shows the quadtree structure that we obtain for the encoding of *cameraman* at a bit-rate of 0.2 bpp. Notice that the reconstructed image (Figure 5.5(c)) has a higher PSNR (about 1dB) than what we obtain using a Jpeg2000 encoder (we use the java implementation of the Jpeg2000 reference software available at: *http://jj2000.epfl.ch*).

### 5.3.2 Our distributed encoding approach for stereo pairs with arbitrary bit-rate allocation

We consider a scene consisting of a set of vertical polygons of linear intensities that are placed at different depths between the $z_{min}$ and $z_{max}$ values (the polygons

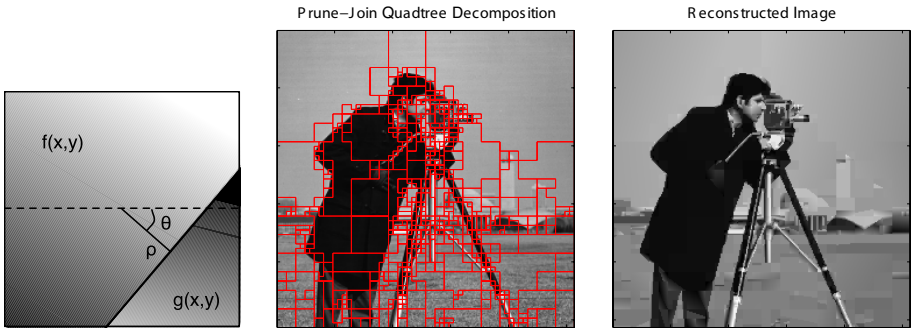


**Figure 5.5: (a) Our geometrical model consists of two 2D linear regions separated by a 1D linear boundary. This boundary is represented with two coefficients ($\theta$,$\rho$) and each 2D region of the tile is represented with three coefficients $(c_1, c_2, c_3)$ such that $f(x,y) = \sum_{i=1}^{3} c_i \mathcal{L}_i(x,y)$, where $\{\mathcal{L}_i(x,y)\}_{i=1}^{3}$ forms an orthonormal basis for 2D linear functions over the region covered by $f(x,y)$. (b) Prune-Join quadtree decomposition of cameraman with a target bit-rate of $0.2$ bpp. (c) The PSNR of the reconstructed image is about 1dB better than what we obtain with a Jpeg2000 encoder.**

**Algorithm 5.2** Prune-Join quadtree encoding algorithm

1: Segmentation of the signal using a quadtree decomposition up to a maximum depth $J_{max}$.
2: Approximation of each node of the quadtree by a geometrical model consisting of two 2D linear pieces separated by a 1D linear boundary (see Figure 5.5(a)).
3: Rate-Distortion curves generation for each node of the quadtree using scalar quantization and optimal bit allocation on the 8 coefficients (2 coefficients for the 1D linear boundary and 3 coefficients for each 2D linear piece). Two or three bits of side information per node are needed to indicate the model used (each tile can be represented with one or two 2D polynomials that can be constant or linear).
4: Optimal pruning of the quadtree for the given operating slope $-\lambda$ according to the following Lagrangian cost based criterion: Prune the four children of a node if:
$\sum_{i=1}^{4}(D_{c_i} + \lambda R_{c_i}) \geq (D_p + \lambda R_p)$.
5: Joint coding of similar neighbouring leaves (or groups of already joint leaves) according to the following criterion: Join the two neighbours if:
$(D_{n_1} + \lambda R_{n_1}) + (D_{n_2} + \lambda R_{n_2}) \geq (D_{n_{Joint}} + \lambda R_{n_{Joint}})$.
Two bits of side information are needed to indicate the direction of the joint neighbour (up, down, left or right).
6: Search for the desired R-D operating slope (update $\lambda$ and go back to 4).

can be horizontally tilted such that their right and left extremities are at different depths). Assume now that $V_1(x, y)$ and $V_2(x, y)$ are two views of this scene obtained from two consecutive cameras. These two views are therefore 2D piecewise linear signals defined over $[0; T] \times [0; T]$. They can be represented using the same set of poynomials[3] but their 1D linear discontinuities are shifted according to the range of possible disparities given by the correlation model.

Our distributed coding approach consists in decomposing each view using the quadtree approach presented in Algorithm 5.2, and then transmitting only partial information from each view. The total information necessary to describe each view can be divided in 3 parts: $R_{Tree}$ is the number of bits necessary to code the pruned quadtree and is equal to the number of nodes in the quadtree, $R_{LeafJointCoding}$ is the number of bits necessary to code the joining information and is equal to the number of leaves in the quadtree plus two bits of side information for each joined leave. Finally, $R_{Leaves}$ is the total number of bits necessary to code the geometrical information of the leaves (2D polynomials, 1D boundaries and model side information).

Our approach can be described as follows (asymmetric case):

- Send the full description of $V_1(x, y)$ from the first encoder.
- Sent only the subtrees of the quadtree structure of $V_2(x, y)$ having a root node at level $J_\Delta = \lceil \log_2(\frac{T}{\Delta_{max} - \Delta_{min} + 1}) \rceil$ along with the joining information and the coefficients representing the 1D boundaries.

Note that a more flexible allocation of the bit-rates between the encoders can be easily obtained by letting each of them send complementary subsets of their polynomials in a way similar to the proposed approach in Section 5.2.3. In order to give

[3]Note that the 2D polynomials can be horizontally contracted or expanded but their representation remains the same as we normalize them according to their support.

the correct intuition about this coding approach, we propose a simple example in Section 5.3.4, where the scene consists of a simple rectangle of constant intensity.

### 5.3.3 Joint reconstruction at the decoder and the matching problem

At the decoder, the information obtained from all the encoders is used along with the known a-priori correlation structure to retrieve all the shifts (disparities) and retrieve the missing information about the segmentation of the signals in a way similar to the one in the 1D case (see Section 5.2.3). The missing polynomial coefficients are then simply copied from the view where they are available. This matching of the different quadtree structures is straightforward in the case where the correlation between the views satisfy our piecewise polynomial model exactly, but becomes more involved in the case of real multi-view images. This is due to the fact that the quadtree decomposition can encode, in a particular view, a discontinuity that does not appear in the other views. In certain cases, the decoder can then make errors when matching the two quadtree structures, which would lead to a poor reconstruction. Moreover, if the scene does not fully satisfy the lambertianity assumption, and because of the noise induced by the acquisition system, the quadtree algorithm may, in many cases, not decompose the different views with the exact same set of polynomials.

In order to fix this problem, we can transmit some extra (redundant) information to help the decoder perform a correct matching of the discontinuities of the different views. When matching the different discontinuities, the algorithm can check that the extra information about polynomials shows consistency with the existing polynomials, therefore reducing the risk of selecting a wrong match. The aim of the matching is to retrieve the most reliable depth map of the scene such that the polynomial intensities of the different regions can be correctly used to reconstruct

the views where this information is not available. The depth map is created by fitting linear pieces between the retrieved depth positions of the matched discontinuities. Each polynomial piece can then easily be shifted and stretched according to the depth map before being copied from one view to another. The coding strategy for distributed encoding of real stereo pairs can therefore be modified as follows: First, both views transmit their full quadtree structures, along with the joining information, the side information about the geometrical model used for each region, and the set of 1-D boundaries. Then, the first encoder transmits its full set of polynomials, while the second one only transmits the most significant bits of its constant coefficients. These most significant bits correspond to the redundant information that has to be transmitted in order to help the decoder perform a correct matching of the quadtree structures.

During the reconstruction, each polynomial piece of the first view is matched with its corresponding region in the other view. This matching is done by scanning all the possible positions on the second view according to the epipolar constraints and selecting the one that presents the best characteristics (similar position and orientation of discontinuity and equivalence in the most significant bits of the constant coefficients). Once this best match is identified, the original region of the first image is warped and copied to the corresponding region on the other view.

In Section 5.3.5, we present some simulation results based on this modified algorithm on real multi-view images.

### 5.3.4 A simple example in an ideal scenario

We consider here a simple example to highlight our quadtree based distributed coding technique. Figure 5.6 shows three views of a scene consisting of a single rectangle of constant intensity placed at a certain depth (in front of a constant

background) such that its displacement from one view to the next one is equal to four pixels. The figure also shows the structure of the pruned quadtree (green dashed lines) along with the joining information (red lines). Using our standard prune-join quadtree encoder, we can see that each of these views can be losslessly encoded using a total of 114 bits. The pruned quadtree represented using its full first three levels, therefore consists of $4^0+4^1+4^2 = 21$ nodes and is represented with 21 bits. The joining information requires one bit per leaf to indicate that the leaf is joined to the next one, and two more bits to indicate the direction of the next leaf. The total number of bits necessary to encode the joining information is therefore equal to $16 + 13 \times 2 = 42$ bits. Finally, 51 bits are used to encode the geometrical representations (16 bits for the 1-D boundaries, 32 bits for the polynomials and 3 bits for the model side information).

We assume in this example that the minimum and maximum disparities are given by: $\{\Delta_{min}; \Delta_{max}\} = \{1; 8\}$. This means that $J_\Delta = 2$ and that the first two level of the quadtree structure (5 bits) need to be transmitted from only one encoder. In Table 5.3, we show different bit allocation examples that lead to a perfect reconstruction of the three views. These examples highlight the arbitrary bit allocation and the bit conservation principle of our coding approach. For example, the last line of Table 5.3 presents an example of bit allocation between the three encoders, where each encoder 1 and 3 transmits 77 bits. These are used to represent part of their quadtree structures (starting at level $J_\Delta = 2$) (16 bits), their joining information (42 bits), their side information about the geometrical models (3 bits) and their 1-D boundaries (16 bits). The encoder 2 transmits a total of 37 bits corresponding to the first two level of the quadtree (5 bits) and the polynomial coefficients (32 bits).

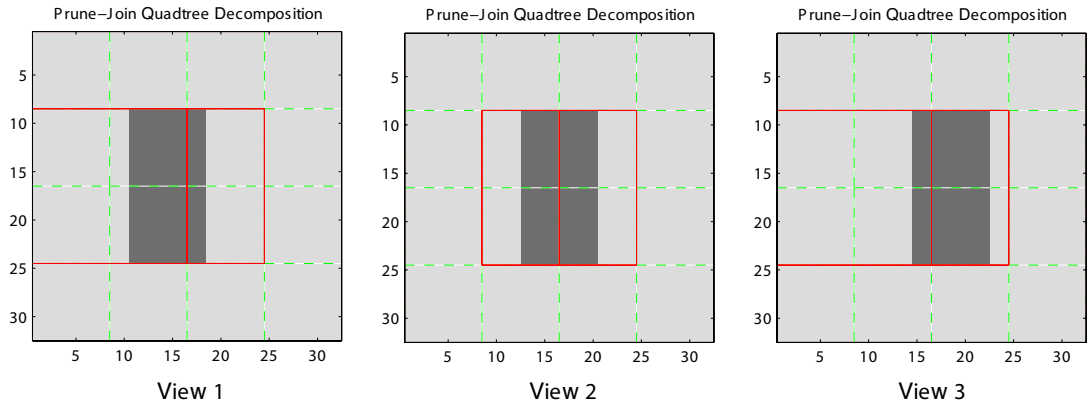


**Figure 5.6: Three $32 \times 32$ views of a very simple synthetic scene consisting of one rectangle at a certain depth. The quadtree decomposition uses a total of $114$ bits to encode each view ($21$ bits for the pruned quadtree, $42$ bits for the joining information and $51$ bits for the geometrical representations).**

**Table 5.3: Numerical results for the lossless encoding of the simple views shown in Figure 5.6**

| **Coding Strategy** | $\mathbf{R_1}$ (bits) | $\mathbf{R_2}$ (bits) | $\mathbf{R_3}$ (bits) | $\mathbf{R_{tot}}$ (bits) |
|---|---|---|---|---|
| Independent | 114 | 114 | 114 | 342 |
| SW asym. - 2 views | 114 | - | 77 | 191 |
| SW sym.- 2 views | 95 | - | 96 | 191 |
| SW asym. - 3 views | 77 | 37 | 77 | 191 |

### 5.3.5 Simulation results on real 2-D multi-view images

We present some simulation results obtained on the *lobby* multi-view sequence from Shum et al. [71]. Figure 5.7 shows the result of an asymmetric encoding of a stereo pair where the first view is encoded at 0.32 bpp, whereas the second view is encoded at a significantly lower bit-rate and some information about its polynomial coefficients is discarded. After the matching, the missing coefficients can be retrieved from the first view, which improves the quality of the reconstructed view.

Figures 5.8 and 5.9 show results obtained on a sequence of six consecutive views, where the first and the sixth views are fully transmitted and only the quadtree structure is transmitted for the other four views. Figure 5.8 shows that our approach outperforms an independent encoding of the six views for all the range of considered bit-rates. In Figures 5.10, 5.11 and 5.12, the difference in the reconstruction quality between the independent and distributed approach is highlighted. The images correspond to the $5^{th}$ view of the lobby sequence.

Since the different views of the scene do not fully satisfy our correlation model, the reconstruction quality of any view for which the original polynomial coefficients have not been transmitted is limited, even when the quality of the other views is increased. This residual error is typically due to the noise, the non-lambertianity of the scene, or the presence of non-flat surfaces. To some extend, the problem of the noise is less problematic for the reconstruction algorithm than the possible geometrical discrepancies between the scene and the correlation model. This is due to the fact that the quadtree decomposition algorithm is naturally robust to the presence of noise and can actually be used as an efficient image denoiser as demonstrated in [70].

Figures 5.13, 5.14, 5.15, 5.16 and 5.17 show other simulation results obtained on different set of multi-view images. Again, the distributed approach outperforms

the independent compression algorithm.


**Figure 5.7: Distributed stereo encoding of two views. View 1 (left) is encoded at $0.32$ bpp and fully transmitted. View 2 (right) is encoded at a much lower bit-rate and some of its polynomial coefficients are discarded. Joint decoding of view 2 (center) shows an improvement of the reconstruction quality of about $1$dB compared to an independent encoding.**

## 5.4 Conclusions

In this chapter, we have shown how our distributed compression approach proposed in Chapter 3 can be used with a real image coder based on tree structured decompositions. We have shown that our approach allows for an arbitrary bit-rate allocation, and we have highlighted an exact bit conservation principle. Our approach has been shown to outperform independent encoding of real multi-view images even when our piecewise polynomial correlation model is not fully respected.

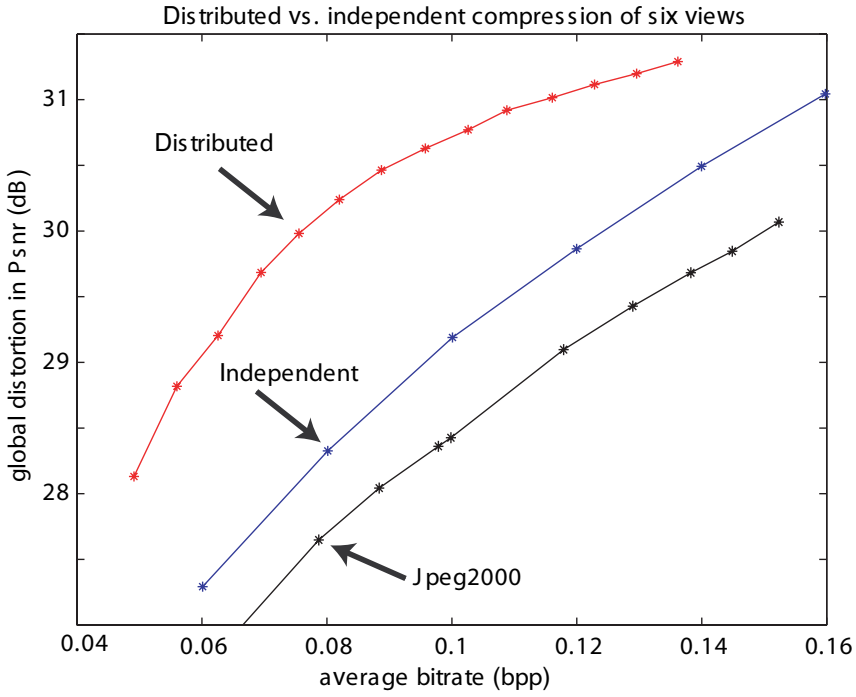


**Figure 5.8: Distributed vs. independent encoding of six views. Results obtained with a Jpeg2000 encoder are also shown.**

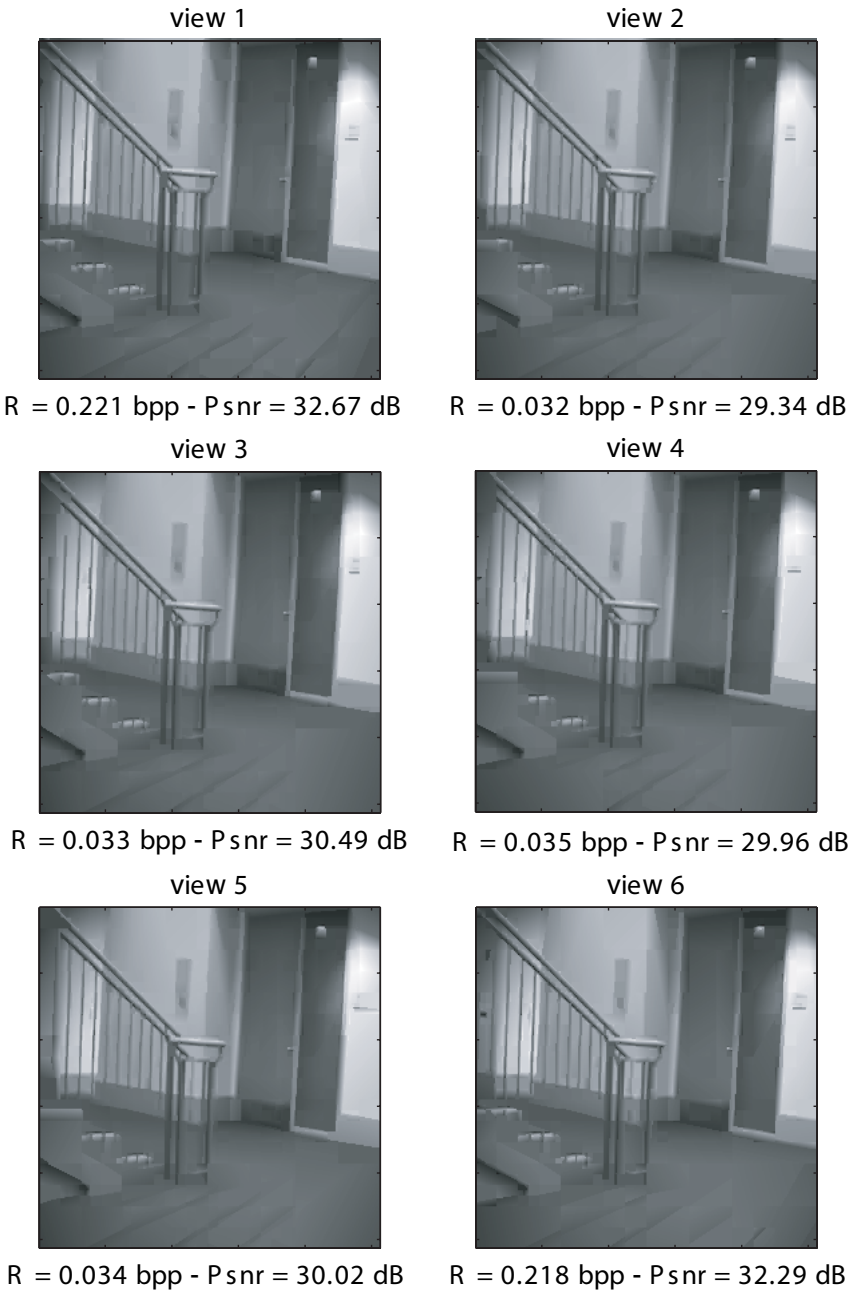


**Figure 5.9: The six images correspond to a distributed encoding with an average bit-rate of $0.08$ bpp. Images 1 and 6 are encoded indepedently while images 2 to 5 are encoded using our distributed compression approach.**

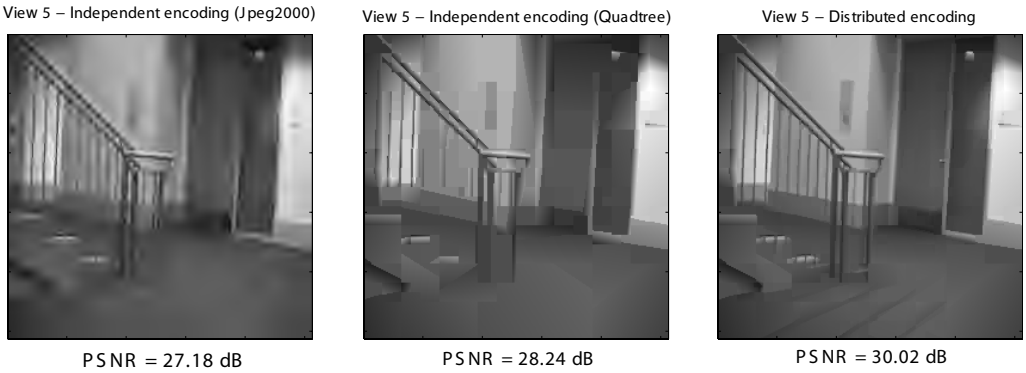


**Figure 5.10: Reconstruction of the $5^{th}$ view. Independent encoding at $0.08$ bpp with Jpeg2000 (left). Independent encoding at $0.08$ bpp with the prune-join quadtree coder (center). Our distributed encoding approach with an average of $0.08$ bpp (right).**

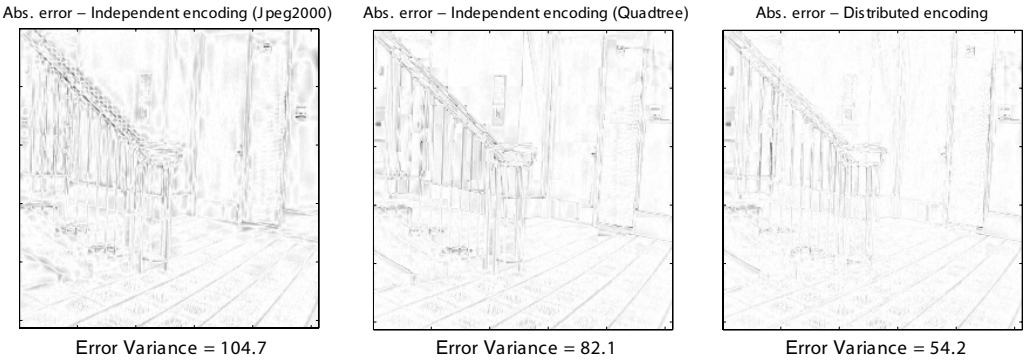


**Figure 5.11: Variance of the errors of reconstruction for the $5^{th}$ view encoded at $0.08$ bpp.**

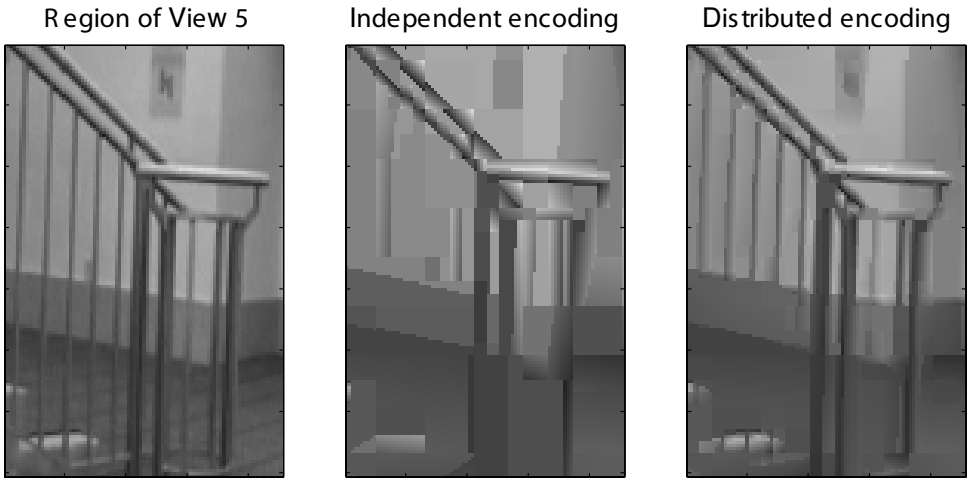


**Figure 5.12: Visual quality of a specific region of the $5^{th}$ view at the decoder. Original quality (left). Independent encoding at $0.08$ bpp (center). Distributed encoding with an average of $0.08$ bpp (right).**

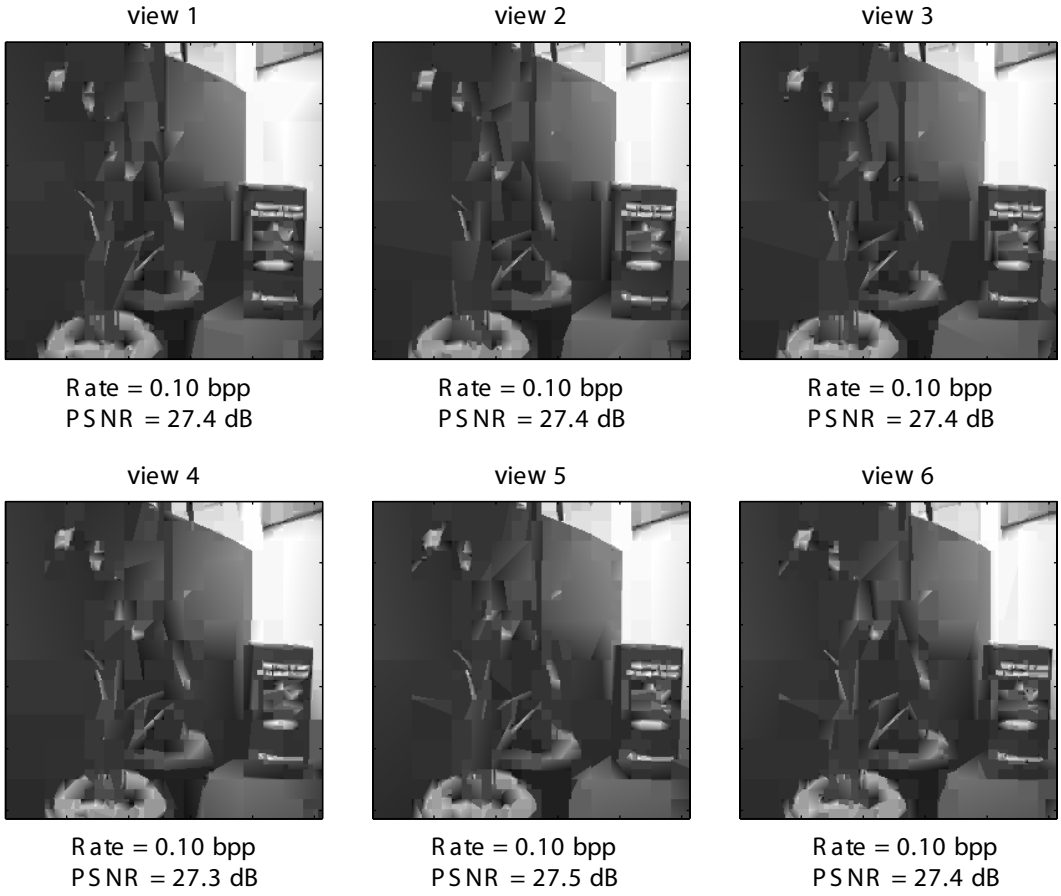


**Figure 5.13: Independent encoding of six views at $0.1$ bpp.**

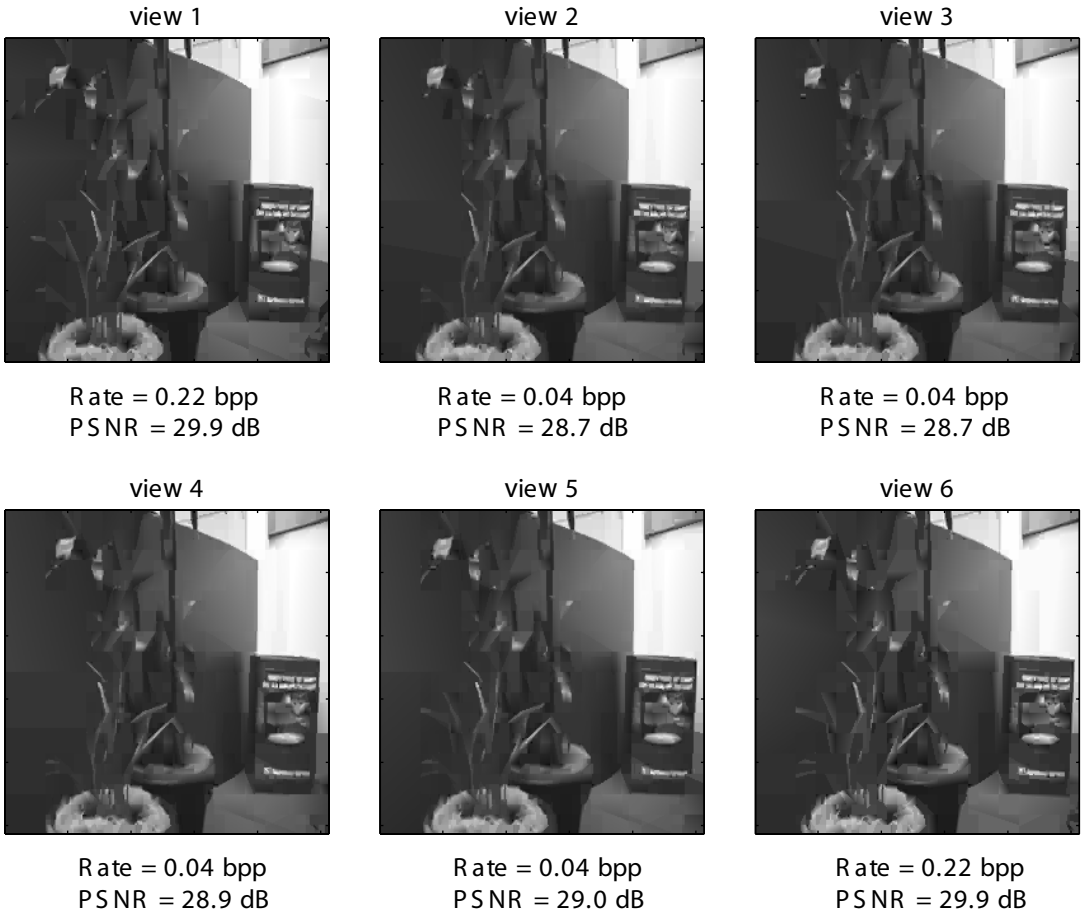


**Figure 5.14: Distributed encoding of the six views with an average bit-rate of $0.1$ bpp. Images 1 and 6 are encoded independently while images 2 to 5 are encoded using our distributed compression approach.**

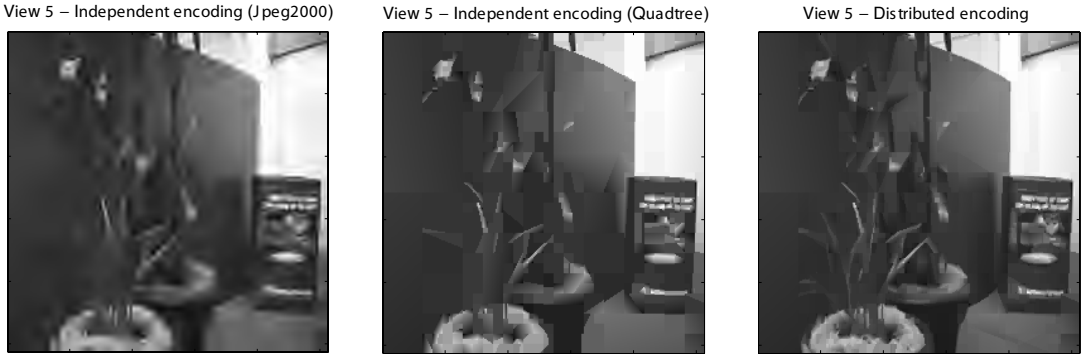


**Figure 5.15: Reconstruction of the $5^{th}$ view. Independent encoding at $0.1$ bpp with Jpeg2000 (left). Independent encoding at $0.1$ bpp with the prune-join quadtree coder (center). Our distributed encoding approach with an average of $0.1$ bpp (right).**

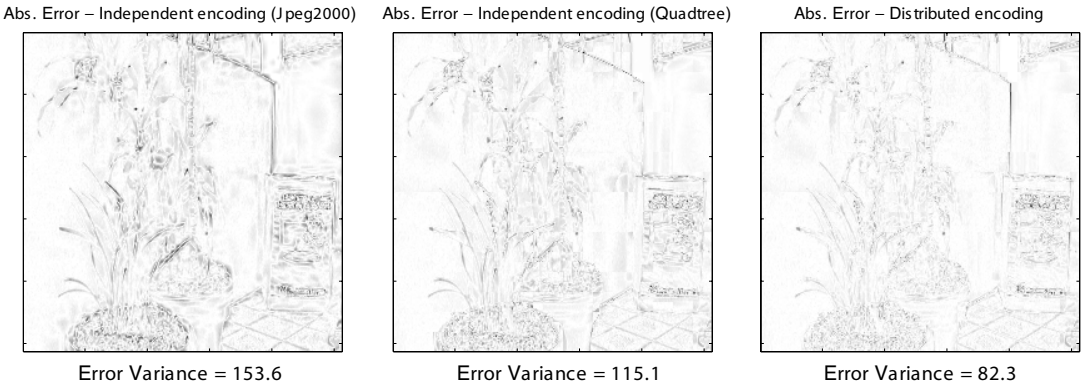


**Figure 5.16: Variance of the errors of reconstruction for the $5^{th}$ view encoded at $0.1$ bpp.**

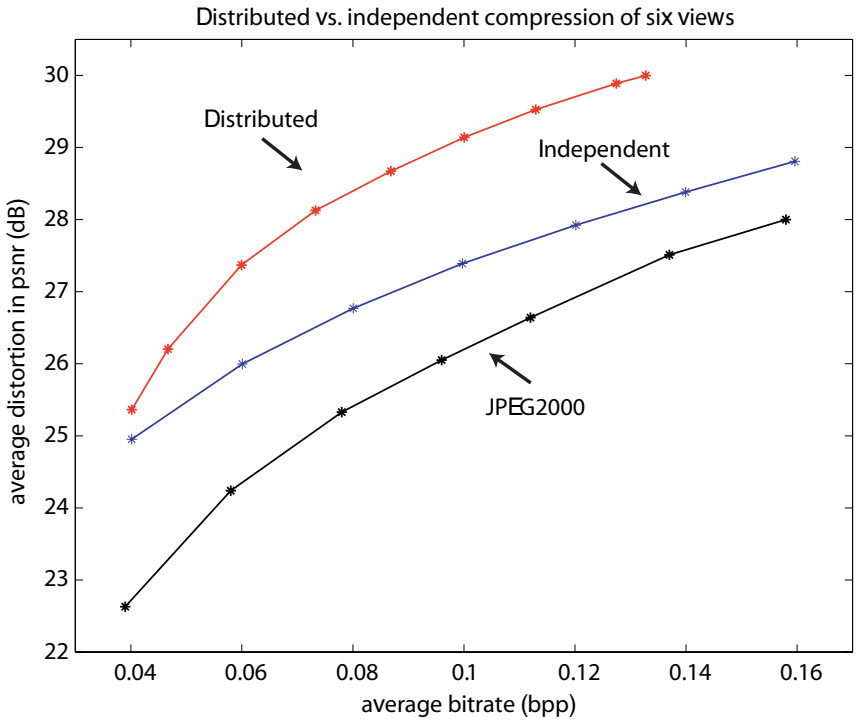


**Figure 5.17: Distributed vs. independent encoding of six views. Results obtained with a Jpeg2000 encoder are also shown.**

# Chapter 6

# Distributed Coding of Binary Sources

## 6.1 Introduction

Most of the distributed source coding approaches introduced in Chapter 2 present very good performances, but mainly focus on the asymmetric scenario, where one of the two sources is transmitted perfectly to the receiver. For practical applications, however, it might be necessary to have more flexibility in the repartition of the bit-rates between the encoders.

In [53], Pradhan and Ramchandran proposed a technique based on their original work (DISCUS [52]) in order to achieve any point of the Slepian-Wolf achievable rate region. Their method creates two sub-codes of a single channel code by splitting the original generator matrix in two. Then, each encoder uses one of these sub-codes to encode its data. Several practical approaches inspired from these results have recently been proposed in [11, 64, 74].

In this chapter, we propose our own constructive approach (inspired from our coding approach for multi-view images presented in Chapter 3) that allows for a

flexible repartition of the transmission rates between the encoders. Our technique uses a single linear channel code that can be non-systematic [59]. The performance of our approach depends only on the quality of the channel code used. As presented in Figure 2.3, the two correlated sources can be seen respectively as the input and the output of a certain channel used to model their correlation. We refer to this virtual channel as the *correlation channel* of the two sources. If we can find a code that achieve the capacity of this *correlation channel*, then our distributed source coding approach can reach any point of the Slepian-Wolf bound and is therefore optimal.

Notice that similar approaches have recently been proposed in [11, 64, 74, 75]. Although our own approach is relatively similar in spirit to the one in [74], our scheme can also be used with non-systematic codes, whereas their technique can only be used with systematic codes. Note that good capacity achieving LDPC codes are usually non-systematic. In [64] and [11], iterative decoding procedures are proposed in order to decode the two correlated blocks simultaneously. Their main strategy is to apply the standard sum-product algorithm (message-passing decoding [60]) on an extended factor graph, corresponding to two standard LDPC decoders connected through correlation nodes modeling the joint distribution between the sources. In Section 6.3, we show that our approach does not require the use of such extended factor graphs since our methods only needs to decode one single block (the difference pattern). A standard iterative decoding scheme similar to the one proposed in [38] can therefore be used in our case. Our approach can thus be seen as an intuitive extension of the asymmetric approach proposed in [38] in order to achieve the entire Slepian-Wolf achievable rate region.

## 6.2 A Simple Example with the Hamming $(7,4)$ Code

By looking at our coding strategy for multi-view images illustrated in Figure 3.4, we can see that the minimum information required at the decoder from both sources is the last $R_{min}$ bits. This information corresponds to the uncertainty of an object's position on an image given its position on the other image. It is therefore related to the conditional entropy of the position and plays the same role as the syndrome in the general case of binary sources. An intuitive generalization of this coding strategy for correlated binary sources would consist in sending the syndrome from each source and then, only complementary subsets of their first $I(x,y)$ bits.

Assume $x$ and $y$ are two uniformly distributed 7-bit binary random variables obtained from two correlated sources such that the Hamming distance between them cannot exceed one ($d_H(x,y) \leq 1$). We consider the Hamming $(7,4)$ code $\mathcal{C}$ with the following parity check matrix:

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{6.1}$$

This code $\mathcal{C}$ is known to have a minimum distance of three and can therefore be used as a channel code able to correct up to one erroneous bit per 7-bit codeword.

Assume that the $x$ and $y$ belong to cosets $i$ and $j$ respectively. The difference between $x$ and $y$ is given by the error pattern $e_k = x \oplus y$ ($x$ and $y$ differs at position $k$), where $\oplus$ corresponds to the binary addition. We know that the syndromes of $x$ and $y$ are given by $s_x = \mathbf{H}x^T = \mathbf{H}e_i^T$ and $s_y = \mathbf{H}y^T = \mathbf{H}e_j^T$ respectively, where $e_i$

and $e_j$ are the error patterns of the cosets $i$ and $j$ respectively. We can observe that:

$$s_k = \mathbf{H}e_k^T = \mathbf{H}(x \oplus y)^T = \mathbf{H}x^T \oplus \mathbf{H}y^T = s_x \oplus s_y. \tag{6.2}$$

This result shows that knowing only the syndromes of $x$ and $y$, we can retrieve the syndrome of their difference pattern and, therefore, the bit position where they differ. We assume the following block representations for $x$, $y$ and $\mathbf{H}$:

$$x = [x_a \quad x_b] \qquad y = [y_a \quad y_b] \qquad \mathbf{H} = [\mathbf{H}_a \quad \mathbf{H}_b] \tag{6.3}$$

where the first and the second blocks are of length four and three respectively. The coding approach proposed in Figure 6.1 consists in sending $[x_1 \quad x_2 \quad s_x^T]$ from encoder 1 and $[y_3 \quad y_4 \quad s_y^T]$ from encoder 2.

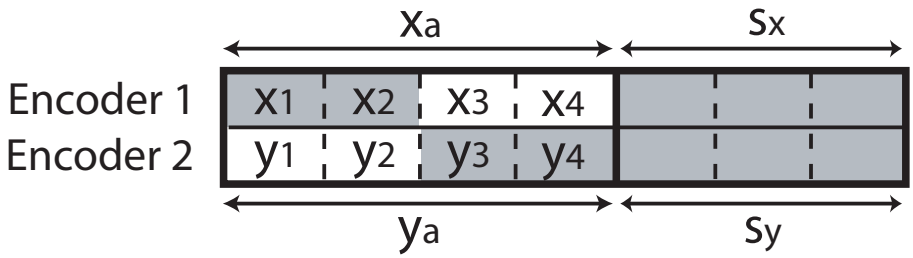


**Figure 6.1: Example of distributed source coding of two correlated 7-bit blocks using the Hamming $(7, 4)$ code. A total of 10 bits are sent to the receiver (gray squares).**

At the decoder, the syndrome of the difference pattern between $x$ and $y$ is obtained by computing the sum of the two syndromes $s_x \oplus s_y$. Using the corresponding error pattern (recovering the difference position), the missing bits of $x_a$ and $y_a$ can easily be retrieved. Finally, $x_b$ and $y_b$ are obtained as: $x_b^T = \mathbf{H}_b^{-1}(s_x \oplus \mathbf{H}_a x_a^T)$ and

$y_b^T = \mathbf{H}_b^{-1}(s_y \oplus \mathbf{H}_a y_a^T)$ where the inverse matrix of $\mathbf{H}_b$ is given by:

$$\mathbf{H}_b^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{6.4}$$

This coding scheme is able to transmit $x$ and $y$ to the decoder using any of the following pair of rates: $(R_1, R_2) \in \{(3,7); (4,6); (5,5); (6,4); (7,3)\}$, and provides therefore more flexibility than the original asymmetric approach. Figure 6.2 highlights a practical example for the symmetric encoding of a given pair $(x, y)$. In the next section, we propose a formal extension in order to use this intuitive approach with any binary linear code.

## 6.3 Constructive Approach using any Linear Channel Code

Assume we have an $(n, k)$ binary linear code $\mathcal{C}$ with parity check matrix $\mathbf{H}$ in its reduced form such that: $\mathbf{H} = [\mathbf{H}_1 \quad \mathbf{H}_2]$, where $\mathbf{H}_2$ is a non-singular square matrix. Notice that if the code is systematic (i.e., the generator matrix is of the form: $\mathbf{G} = [\mathbf{I}_k \quad \mathbf{P}]$), then $\mathbf{H}$ can be given by: $\mathbf{H} = [-\mathbf{P}^T \quad \mathbf{I}_{n-k}]$. In this case, $\mathbf{H}_2$ is simply the identity matrix.

We know that such a code is capable of correcting $2^{n-k}$ different error patterns. In other words, this code $\mathcal{C}$ generates $2^{n-k}$ cosets each containing $2^k$ codewords of length $n$. Let $x_i$ be a binary block of length $n$ represented as: $x_i = [a_i \quad b_i \quad q_i]$, where $a_i, b_i$ and $q_i$ are of length $k_1, k_2$ and $n-k$ respectively ($k_1$ and $k_2$ are chosen such that their sum is equal to $k$ and determine how the total bit-rate will be distributed between the two encoders). The syndrome of $x_i$ is defined as: $s_i =$

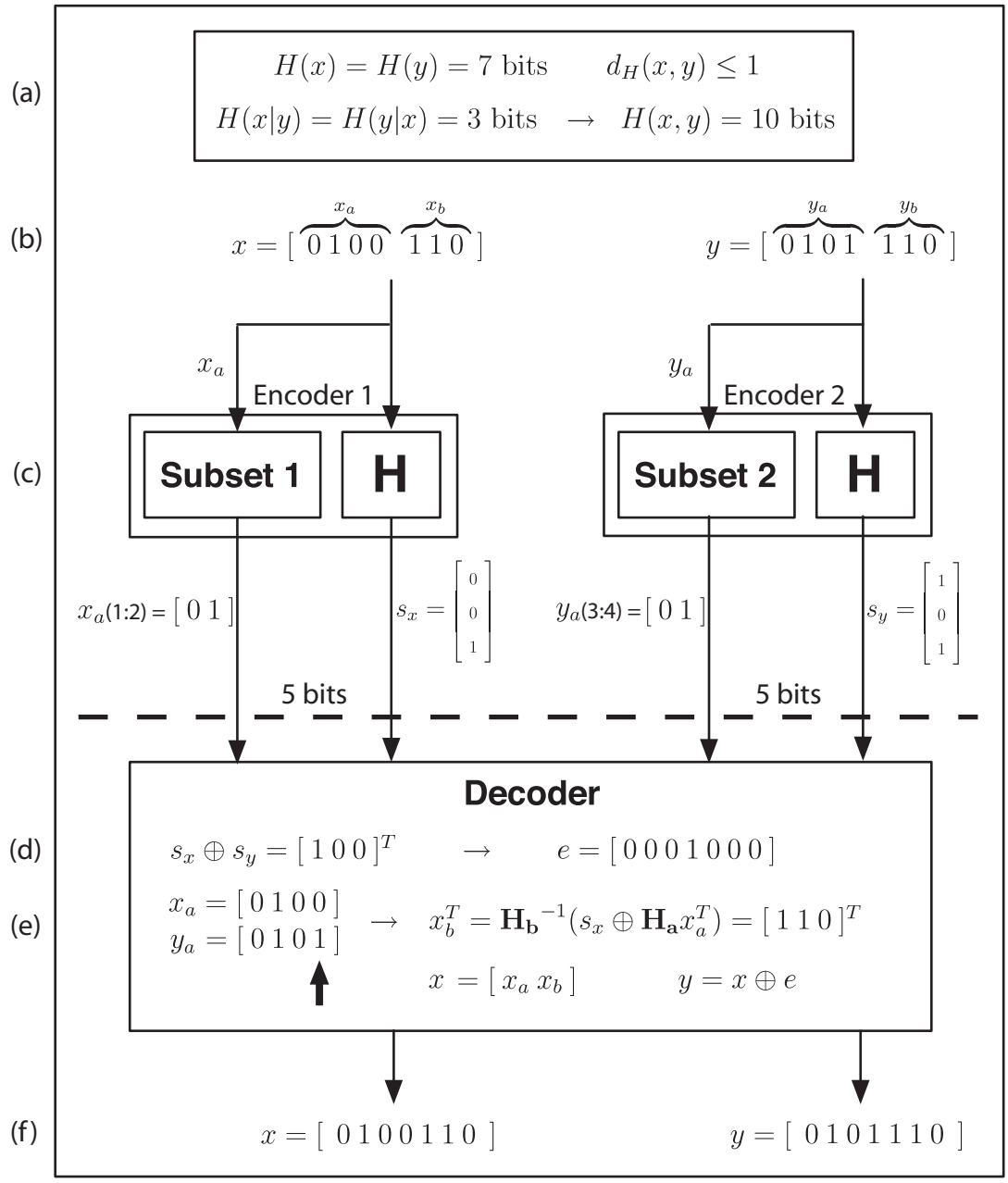


**Figure 6.2: Practical example of symmetric encoding of two 7-bit blocks $x$ and $y$. (a) $x$ and $y$ are uniformly distributed and they are correlated such that their Hamming distance is at most one. (b) The two realisations for $x$ and $y$. (c) The encoders transmit complementary subsets of the first 4 bits of their blocks along with the syndromes. (d) The decoder retrieves the error pattern by adding the two syndromes. (e) The missing bits from $x_a$ and $y_a$ are recovered. Then, $x_b$ is computed and $y$ is simply obtained by adding the error pattern to $x$. (f) $x$ and $y$ are perfectly recovered by the decoder for a total transmission of 10 bits that corresponds to $H(x,y)$. The coding is therefore optimal.**

$\mathbf{H}_1[a_i \quad b_i]^T \oplus \mathbf{H}_2 q_i^T$. We know that $x_i$ belongs to the coset number $k$, if its syndrome is given by: $s_i = \mathbf{H}x_i^T = \mathbf{H}e_k^T$ ($e_k$ is the coset leader of coset number $k$, i.e., the codeword with minimum weight).

Consider now two $n$-bit blocks $x_1$ and $x_2$, correlated such that their Hamming distance $d_H(x_1, x_2)$ is at most $m$. Assume that the channel code $\mathcal{C}$ is able to correct up to $M \geq m$ errors. We know that the following relation must hold:

$$2^{n-k} \geq \sum_{j=0}^{M} \binom{n}{j} \qquad \text{(sphere packing bound)}. \tag{6.5}$$

Our distributed coding strategy consists in sending $[a_1 \quad s_1^T]$ and $[b_2 \quad s_2^T]$ from the encoders 1 and 2 respectively. The transmission bit-rates are therefore given by: $R_1 = n - k_2$ bits and $R_2 = n - k_1$ bits, corresponding to a total of $R_1 + R_2 = 2n - k$ bits.

At the receiver, we let $e_d$ correspond to the "difference pattern" between $x_1$ and $x_2$ as: $e_d = x_1 \oplus x_2$. We know that the syndrome of $e_d$ is given by $s_d = \mathbf{H}e_d^T = \mathbf{H}(x_1^T \oplus x_2^T) = s_1 \oplus s_2$. We can now retrieve the error pattern $e_d$ corresponding to this syndrome $s_d$ using one of the following techniques: If the code is not too large, a simple lookup table storing the corresponding pattern error for each possible syndrome can be used. For larger code, an iterative method has to be used. Using an iterative decoding scheme such as the one proposed in [38], we can recover $e_d$ as the closest codeword to the all zero sequence satisfying the syndrome $s_d$. Notice that this iterative decoding approach is particularly suited for LDPC codes which are amongst the best block codes known for memoryless channels.

Knowing the difference pattern $e_d$, the missing bits of the $k$ first bits of $x_1$ and $x_2$ are easily obtained as: $[a_2 \quad b_1] = [a_1 \quad b_2] \oplus e_d^k$, where $e_d^k$ corresponds to the $k$ first bits of $e_d$.

The syndrome of $x_1$ is given by: $s_1 = \mathbf{H}_1[a_1 \quad b_1]^T \oplus \mathbf{H}_2 q_1^T$. Let $z_1$ be defined as:

$z_1 = s_1 \oplus \mathbf{H}_1[a_1 \quad b_1]^T$. We can now retrieve $q_1$ by computing: $q_1^T = \mathbf{H}_2^{-1} z_1$. Notice that $\mathbf{H}_2^{-1}$ can be obtained using Gaussian Elimination and that, if $\mathcal{C}$ is systematic, we can have $\mathbf{H}_2 = \mathbf{I}_{n-k}$ and $q_1 = z_1^T$. Knowing $q_1$, we have now completely recovered $x_1$ and we can easily obtain $x_2$ as $x_2 = x_1 \oplus e_d$.

In terms of performance, we can say that the ability of our distributed source coding technique to work close to the Slepian-Wolf bound only depends on the quality of the channel code used. More specifically, if $x_1$ and $x_2$ are uniformly distributed and have a joint distribution $p(x_1, x_2)$, then the closer the channel code $\mathcal{C}$ gets to the capacity of the virtual binary channel defined by $p(x_2|x_1)$, the closer our system gets to the Slepian-Wolf bound. The design of capacity achieving channel codes, however, is beyond the scope of this thesis.

## 6.4 Generalization to more than two Sources

Our approach presented in the previous section can be extended to any number of correlated sources. In particular, consider $N$ correlated binary sources producing binary blocks of length $n$ $(x_1, x_2, \ldots, x_N)$, such that the Hamming distance of two consecutive sequences is at most $m$ (i.e., $d_H(x_i, x_{i+1}) \leq m$ for $i = 1, \ldots, N-1$). Our coding strategy proposed in Section 6.3 can be extended to this $N$ sources scenario (see Figure 6.3) through the following proposition:

**Proposition 6.1.** *Assume $x_1, \ldots, x_N$ are $N$ binary sequences of length $n$ correlated such that the Hamming distance between two consecutive sequences is at most $m$ (i.e., $d_H(x_i, x_{i+1}) \leq m$ for $i = 1, \ldots, N-1$). Consider an $(n, k)$ linear channel code $\mathcal{C}$ that can correct perfectly up to $M \geq m$ errors per $n$-bit block. The following distributed coding strategy uses a total of $n + (N-1)(n-k)$ bits to encode the $N$ sequences and is sufficient to allow for a perfect reconstruction of all of them at the decoder:*

- *From each encoder, send the syndrome $s_i$ of the corresponding block $x_i$.*
- *Send only complementary subsets of their first k bits such that each bit position is sent from only one encoder.*

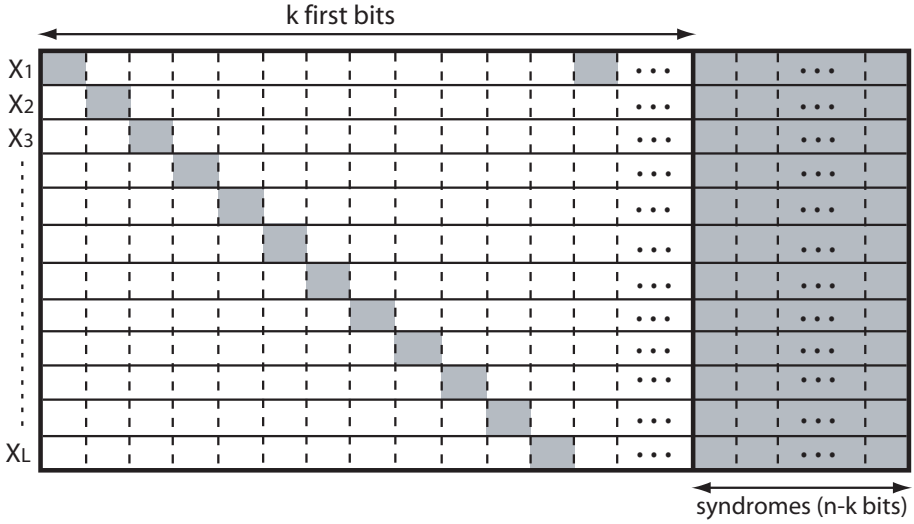


**Figure 6.3: Our encoding strategy for $N$ correlated binary sources. Each encoder sends the syndrome and a subset of the first $k$ bits of their input block. The subsets are chosen such that each bit position is sent from only one source.**

At the decoder, the $N-1$ difference patterns can be recovered from the $N$ syndromes, allowing then to complete the first $k$ bits of each sequence. The decoding method used here is similar to the one presented in the previous section. The difference pattern of each pair of consecutive blocks is retrieved by running the standard iterative decoding method with the sum of the two syndromes. Knowing all the difference patterns and having received complementary subsets of the first $k$ bits, the first $k$ bits of each block can then be recovered. Finally, each original block $x_i := [a_i \quad q_i]$ $(i = 1, \ldots, N)$ is completed by recovering its last $n-k$ bits as:

$$q_i^T = \mathbf{H}_2^{-1}(s_i \oplus \mathbf{H}_1 a_i^T). \tag{6.6}$$

This coding strategy is in some cases optimal. For instance, if the $N$ sources

$(x_1, x_2, \ldots, x_N)$ are uniformly distributed, the Hamming distance of two consecutive sequences is at most $m$ and the sequences $x_1, x_2, \ldots, x_N$ form a Markov chain; then, if the code $\mathcal{C}$ is such that $M = m$ and Equation (6.5) is satisfied with equality (i.e., $\mathcal{C}$ is perfect), we can write:

$$\begin{aligned} H(x_1, x_2, \ldots, x_N) &= H(x_1) + \sum_{i=2}^{N} H(x_i|x_{i-1}) && (6.7) \\ &= n + (N-1)(n-k). && (6.8) \end{aligned}$$

Our scheme, as indicated in Proposition 6.1, uses $n + (N-1)(n-k)$ bits and is therefore optimal in this case.

## 6.5 Conclusions

In this Chapter, we have proposed an intuitive distributed source coding technique for general correlated binary sources inspired from our coding approach for multi-view images presented in Chapter 3. Our approach can use any linear code (systematic or not), and allows for a flexible bit-allocation between the two encoders. It can therefore achieve any point of the Slepian-Wolf region. We have also shown that this method can be easily extended to the case of more than two sources.

# Chapter 7

# Conclusions

## 7.1 Summary

### Distributed compression of the plenoptic function

We have proposed a distributed compression strategy for simple synthetic scenes that can truly exploit the geometrical correlation available in multi-view images. We have introduced the plenoptic function which can be used to represent the geometrical dependencies between the different views. Then, we have proposed a coding scheme that can be used to efficiently encode the positions of the objects on the different views in a fully distributed manner. Finally, we have shown that our approach can be made resilient to the problem of visual occlusions and have presented simulation results.

### Fundamental trade-offs in camera sensor networks

We have studied some fundamental trade-offs in camera sensor networks. In particular, we have shown that if the observed scene is of finite rate of innovation (FRI) and can be represented exactly with polygons of polynomial intensities, an exact sam-

pling strategy can be devised. Moreover, we have derived rate-distortion bounds for the reconstruction of the views at the decoder and we have shown that an exact bit-conservation principle exists. In other words, the quality of the reconstructed views only depends on the total transmission bit-rate and not on the number of sensors involved in the encoding process.

**Distributed compression of real multi-view images**

We have proposed a practical algorithm for distributed compression of real multi-view images. We have presented the piecewise polynomial model that we use to represent the different views and have shown that a tree-based coding approach recently proposed in the literature can be extended to take advantage of our distributed coding scheme and is optimal in idealized scenarios. We have proposed a detailed description of our distributed algorithm for the 1-D case using binary tree segmentation, and then have shown how to extend it to the 2-D case using a quadtree approach. Finally, we have presented some simulation results obtained on real multi-view images.

**Distributed source coding of binary sources**

We have proposed a distributed source coding approach for correlated binary sources directly inspired from the scheme presented for the distributed encoding of multi-view data. In particular, we have shown that this intuitive approach can cover the entire Slepian-Wolf achievable rate region, and have presented a simple example to give the correct intuition.

## 7.2 Future Research

### Extension to other multi-camera configurations

In our work, we have focused on linear camera arrays. This simple scenario has allowed us to precisely derive the correlation structure between the different views and obtain exact results for the problem of distributed compression of this particular multi-view information. In future works, it would be of interest to consider other geometrical multi-camera set-ups such as cameras on a circle, on a cube, on a sphere or in any other configuration. For each of these configurations, the structure of the acquired data can be estimated by observing the corresponding constraints imposed on the plenoptic function. The distributed compression approach that we proposed in this thesis could then be adapted to this new data structure.

### Better modelling of the prediction error

The piecewise-polynomial-based correlation model that we proposed in our work has the main advantage of being relatively simple but can sometimes fail to accurately represent the real correlation in multi-view data. In particular, the lambertianity and occlusion-free assumptions are rarely satisfied in real scenarios. In future work, the correlation model should therefore be improved to consider a more realistic structure. For example, the polynomial intensity of an object on different views can be slightly different due to reflections. These differences could be modelled to improve the global correlation structure.

### Distributed video compression

The distributed multi-view coding scheme presented in our work focuses on the compression of static scenes. This coding approach could be extended to perform

distributed video compression of single video sources (see Section 2.3.2 for a description of existing DVC approaches). The multi-view correlation could be replaced by a inter-frame correlation model, where the constraints on the disparities are replaced by constraints on the motion of the objects (the min and max disparities in static multi-view could typically be replaced by min and max speeds in video).

### Distributed multi-view video compression

The distributed static multi-view coding approach and the distributed video coding approach could lead to the development of a distributed multi-view video coding approach. The analysis of the global correlation structure is more involved in this context but could be estimated using some constraints on the position of the cameras, the geometry of the scene and the motion of the objects.

# Bibliography

[1] A. Aaron and B. Girod. Compression with side information using Turbo codes. In *Data Compression Conference (DCC '02)*, April 2002.

[2] A. Aaron, P. Ramanathan, and B. Girod. Wyner-Ziv coding of light fields for random access. In *IEEE International Workshop on Multimedia Signal Processing (MMSP '04)*, September 2004.

[3] E.H. Adelson and J.R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J.A. Movshon, editors, *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, Massachusetts, 1991.

[4] I.F. Akyildiz, Su Weilian, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[5] S. Appadwedula, V.V. Veeravalli, and D.L. Jones. Energy-efficient detection in sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):693–702, April 2005.

[6] H. Aydinoglu and M.H. Hayes. Stereo image coding: a projection approach. *IEEE Transactions on Image Processing*, 7(4):506–516, April 1998.

[7] B. Bai, P. Boulanger, and J. Harms. An efficient multiview video compression scheme. In *IEEE International Conference on Multimedia and Expo*, pages 836–839, July 2005.

[8] N.V. Boulgouris and M.G. Strintzis. A family of wavelet-based stereo image coders. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(10):898–903, October 2002.

[9] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 307–318. ACM Press/Addison-Wesley Publishing Co., 2000.

[10] A. Chebira, P. L. Dragotti, L. Sbaiz, and M. Vetterli. Sampling and interpolation of the plenoptic function. In *IEEE International Conference on Image Processing (ICIP '03)*, September 2003.

[11] T.P. Coleman, A. H. Lee, M. Medard, and M. Effros. On some new approaches to practical Slepian-Wolf compression inspired by channel coding. In *Data Compression Conference (DCC '04)*, March 2004.

[12] T.P. Coleman, M. Medard, and M. Effros. Time-sharing vs. source-splitting in the Slepian-Wolf problem: Error exponents analysis. In *Data Compression Conference (DCC '06)*, March 2006.

[13] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.

[14] I. Csiszar. Linear codes for sources and source networks: Error exponents, universal coding. *IEEE Transactions on Information Theory*, 28(4):585–592, July 1982.

[15] P.L. Dragotti, M. Vetterli, and T. Blu. Exact sampling results for signals with finite rate of innovation using strang-fix conditions and local kernels. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP '05)*, volume 4, pages 233–236, March 2005.

[16] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. The MIT Press, Cambridge, MA, USA, 2001.

[17] M. Flierl and P. Vandergheynst. Distributed coding of highly correlated image sequences with motion-compensated temporal wavelets. *EURASIP Journal on Applied Signal Processing, Special Issue on Video Analysis and Coding for Robust Transmission*, 2006(id 46747):1–10, 2006.

[18] T.J. Flynn and R.M. Gray. Encoding of correlated observations. *IEEE Transactions on Information Theory*, 33(6):773–787, November 1987.

[19] J. Garcia-Frias. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, 5(10):417–419, October 2001.

[20] J. Garcia-Frias and F. Cabarcas. Approaching the Slepian-Wolf boundary using practical channel codes. In *IEEE International Symposium on Information Theory (ISIT '04)*, June 2004.

[21] M. Gaubatz, A. Vosoughi, A. Scaglione, and S.S. Hemami. Efficient, low complexity encoding of multiple, blurred noisy downsampled images via distributed source coding principles. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP '06)*, May 2006.

[22] N. Gehrig and P.L. Dragotti. DIFFERENT - DIstributed and Fully Flexible image EncodeRs for camEra sensor NeTworks. In *IEEE Int. Conf. on Image Processing (ICIP '05)*, September 2005.

[23] N. Gehrig and P.L. Dragotti. Symmetric and asymmetric Slepian-Wolf codes with systematic and non-systematic linear codes. *IEEE Communications Letters*, 93(1):61–63, January 2005.

[24] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, January 2005. Invited Paper.

[25] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. In *SIGGRAPH, Computer Graphics Proceedings*. ACM Press, May 1996.

[26] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li. Distributed multi-view video coding. In *Proceedings of SPIE Conference on Visual Communications and Image Processing (VCIP' 06)*, January 2006.

[27] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[28] Y.W. Hong, A. Scaglione, R. Manohar, and B. Sirkeci Mergen. Dense sensor networks are also energy efficient: When 'more' is 'less'. In *MILCOM 2005*, October 2005.

[29] P. Ishwar, A. Kumar, and K. Ramchandran. Distributed sampling for dense sensor networks: a "bit-conservation principle". In *Information Processing in Sensor Networks (IPSN '03)*, April 2003.

[30] A. Jagmohan, A. Sehgal, and N. Ahuja. Compression of lightfield rendered images using coset codes. In *37th Asilomar Conference on Signals, Systems, and Computers: Special Session on Distributed Coding*, November 2003.

[31] A. Kashyap, L.A. Lastras-Montano, C. Xia, and Liu Zhen. Distributed source coding in dense sensor networks. In *Data Compression Conference (DCC '05)*, March 2005.

[32] P. Kulkarni, D. Ganesan, and P. Shenoy. The case for multi-tier camera sensor networks. In *International Workshop on Network and Operating Systems Support for Digital Audio and Video (ACM NOSSDAV'05)*, pages 141–146, June 2005.

[33] K. Lajnef, C. Guillemot, and P. Siohan. Wyner-Ziv coding of three correlated Gaussian sources using punctured turbo codes. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '05)*, December 2005.

[34] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH, Computer Graphics Proceedings*. ACM Press, May 1996.

[35] J. Li, Z. Tu, and R.S. Blum. Slepian-Wolf coding for nonuniform sources using turbo codes. In *Data Compression Conference (DCC '04)*, pages 312–321, March 2004.

[36] H. Lin, L. Yunhai, and Y. Qingdong. A distributed source coding for dense camera array. In *IEEE International Conference on Signal Processing (ICSP '04)*, September 2004.

[37] A.D. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades. Slepian-Wolf coding of three binary sources using LDPC codes. In *3rd International Symposium on Turbo Codes and Related Topics*, September 2003.

[38] A.D. Liveris, Z. Xiong, and C.N. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6(10):440–442, October 2002.

[39] A.D. Liveris, Z. Xiong, and C.N. Georghiades. A distributed source coding technique for correlated images using turbo-codes. *IEEE Communications Letters*, 6(9):379–381, September 2002.

[40] A.D. Liveris, Z. Xiong, and C.N. Georghiades. Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes. In *Data Compression Conference (DCC '03)*, March 2003.

[41] A.D. Liveris, Zixiang Xiong, and C.N. Georghiades. Joint source-channel coding of binary sources with side information at the decoder using IRA codes. In *IEEE Workshop on Multimedia Signal Processing (MMSP '02)*, December 2002.

[42] M. Magnor, P. Ramanathan, and B Girod. Multi-view coding for image-based rendering using 3-d scene geometry. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1092–1106, November 2003.

[43] M. Maitre, C. Guillemot, and L. Morin. 3-D model-based frame interpolation for distributed video coding of static scenes. *IEEE Transactions on Image Processing*, 16(5):1246–1257, May 2007.

[44] I. Maravic and M. Vetterli. Exact sampling results for some classes of parametric non-bandlimited 2-d signals. *IEEE Transactions on Signal Processing*, 52(1):175–189, January 2004.

[45] D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff. On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data. In *Information Processing in Sensor Networks (IPSN '03)*, April 2003.

[46] D. Marco and D. Neuhoff. Reliability vs. efficiency in distributed source coding for field-gathering sensor networks. In *Information Processing in Sensor Networks (IPSN '04)*, April 2004.

[47] D. Marpe, T. Wiegand, and G.J. Sullivan. The H.264/MPEG4 advanced video coding standard and its applications. *IEEE Communications Magazine*, 44(8):134–143, August 2006.

[48] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun. Extensions of H.264/AVC for multiview video compression. In *IEEE Int. Conf. on Image Processing (ICIP '06)*, pages 2981–2984, October 2006.

[49] D.L. Neuhoff and S.S. Pradhan. An upper bound to the rate of ideal distributed lossy source coding of densely sampled data. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP '06)*, May 2006.

[50] Y. Oohama. The rate-distortion function for the quadratic gaussian CEO problem. *IEEE Transactions on Information Theory*, 44(3):1057–1070, May 1998.

[51] M.G. Perkins. Data compression of stereopairs. *IEEE Transactions on Communications*, 40(4):684–696, April 1992.

[52] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. In *Data Compression Conference (DCC '99)*, March 1999.

[53] S.S. Pradhan and K. Ramchandran. Distributed source coding: Symmetric rates and applications to sensor networks. In *Data Compression Conference (DCC '00)*, March 2000.

[54] S.S. Pradhan and K. Ramchandran. Group-theoretic construction and analysis of generalized coset codes for symmetric/asymmetric distributed source coding. In *Conference on Information Sciences and Systems (CISS '00)*, March 2000.

[55] P. Prandoni and M. Vetterli. Approximation and compression of piecewise smooth functions. *Phil. Trans. R.Soc.Lond. 1999.*, 357(1760):2573–2591, September 1999.

[56] R. Puri, A. Majumbar, P. Ishwar, and K. Ramchandran. Distributed video coding in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):94–106, July 2006.

[57] R. Puri and K. Ramchandran. PRISM: A video coding architecture based on distributed compression principles. In *40th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.

[58] S. Rane and B. Girod. Analysis of error-resilient video transmission based on systematic source-channel coding. In *Picture Coding Symposium (PCS '04)*, December 2004.

[59] M.Y. Rhee. *Error Correcting Coding Theory*. McGraw-Hill Inc, 1989.

[60] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, February 2001.

[61] B. Rimoldi and R. Urbanke. Asynchronous Slepian-Wolf coding via source-splitting. In *IEEE International Symposium on Information Theory (ISIT '97)*, June 1997.

[62] M. Sartipi and F. Fekri. Distributed source coding in wireless sensor networks using LDPC codes: a non-uniform framework. In *Data Compression Conference (DCC '05)*, page 477, March 2005.

[63] D. Schonberg, S. S. Pradhan, and K. Ramchandran. LDPC codes can approach the Slepian-Wolf bound for general binary sources. In *40th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.

[64] D. Schonberg, S. S. Pradhan, and K. Ramchandran. Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources. In *Data Compression Conference (DCC '04)*, March 2004.

[65] A. Sehgal and N. Ahuja. Robust predictive coding and the Wyner-Ziv problem. In *Data Compression Conference (DCC '03)*, March 2003.

[66] A. Sehgal, A. Jagmohan, and N. Ahuja. Wyner-Ziv coding of video: An error-resilient compression framework. *IEEE Transactions on Multimedia*, 6(2):249–258, April 2004.

[67] S. Shamai, S. Verdu, and R. Zamir. Systematic lossy source/channel coding. *IEEE Transactions on Information Theory*, 44(2):564–579, March 1998.

[68] P. Shukla and P.L. Dragotti. Sampling schemes for 2-d signals with finite rate of innovation using kernels that reproduce polynomials. In *IEEE Int. Conf. on Image Processing (ICIP '05)*, September 2005.

[69] R. Shukla, P.L. Dragotti, M.N. Do, and M. Vetterli. Rate-distortion optimized tree structured compression algorithms for piecewise polynomial images. *IEEE Transactions on Image Processing*, 14(3):343–359, March 2005.

[70] R. Shukla and M. Vetterli. Geometrical image denoising using quadtree segmentation. In *IEEE International Conference on Image Processing (ICIP'04)*, October 2004.

[71] H.-Y. Shum, S.B. Kang, and S.-C. Chan. Survey of image-based representations and compression techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1020–1037, November 2003.

[72] D. Slepian and J.K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

[73] B. Song, O. Bursalioglu, A.K. Roy-Chowdhury, and E. Tuncel. Towards a multi-terminal video compression algorithm using epipolar geometry. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP '06)*, May 2006.

[74] V. Stankovic, A.D. Liveris, Z. Xiong, and C.N. Georghiades. Design of Slepian-Wolf codes by channel code partitioning. In *Data Compression Conference (DCC '04)*, March 2004.

[75] V. Stankovic, A.D. Liveris, Z. Xiong, and C.N. Georghiades. On code design for the Slepian-Wolf problem and lossless multiterminal networks. *IEEE Transactions on Information Theory*, 52(4):1495–1507, April 2006.

[76] G. Taubin, W.P. Horn, F. Lazarus, and J. Rossignac. Geometry coding and VRML. *Proceedings of the IEEE*, 86(6):1228–1243, June 1998.

[77] M. P. Tehrani, T. Fujii, and M. Tanimoto. Distributed source coding of multiview images. In *Proceedings of SPIE Conference on Visual Communications and Image Processing (VCIP' 04)*, January 2004.

[78] M.P. Tehrani, T. Fujii, and M. Tanimoto. The adaptive distributed source coding of multi-view images in camera sensor networks. *IEICE Transactions on Fundamentals*, E88-A(10):2835–2843, October 2005.

[79] G. Toffetti, M. Tagliasacchi, M. Marcon, A. Sarti, S. Tubaro, and K. Ramchandran. Image compression in a multi-camera system based on a distributed source coding approach. In *European Signal Processing Conference (EUSIPCO '05)*, September 2005.

[80] M. Vetterli, P. Marziliano, and T. Blu. Sampling signals with finite rate of innovation. *IEEE Transactions on Signal Processing*, 50(6):1417–1428, June 2002.

[81] R. Wagner, R. Baraniuk, and R. Nowak. Distributed image compression for sensor networks using correspondence analysis and super-resolution. In *IEEE International Conference on Image Processing (ICIP '03)*, September 2003.

[82] M. Wu, A. Vetro, and C.W. Chen. Multiple description image coding with distributed source coding and side information. In *Proceedings of SPIE Conference on Multimedia Systems and Applications VII*, October 2004.

[83] A.D. Wyner. Recent results in the Shannon theory. *IEEE Transactions on Information Theory*, 20(1):2–10, January 1974.

[84] A.D. Wyner. On source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 21(3):294–300, May 1975.

[85] A.D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976.

[86] J.-J. Xiao, A. Ribeiro, Z.Q. Luo, and G.B. Giannakis. Distributed compression-estimation using wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):27–41, July 2006.

[87] Z. Xiong, A.D. Liveris, S. Cheng, and Z. Liu. Nested quantization and Slepian-Wolf coding: A Wyner-Ziv coding paradigm for i.i.d. sources. In *IEEE Workshop on Statistical Signal Processing (SSP '03)*, September 2003.

[88] Z. Xiong, A.D. Liveris, and S.Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, 21(5):80–94, September 2004.

[89] Q. Xu, V. Stankovic, and Z. Xiong. Wyner-Ziv video compression and fountain codes for receiver-driven layered multicast. In *Picture Coding Symposium (PCS '04)*, December 2004.

[90] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao. Wyner-Ziv coding based on TCQ and LDPC codes. In *37th Asilomar Conference on Signals, Systems, and Computers: Special Session on Distributed Methods in Image and Video Coding*, November 2003.

[91] C. Yeo and K. Ramchandran. Robust distributed multi-view video compression for wireless camera networks. In *SPIE Visual Communications and Image Processing (VCIP'07)*, 2007.

[92] R. Zamir. The rate loss in the Wyner-Ziv problem. *IEEE Transactions on Information Theory*, 42(6):2073–2084, November 1996.

[93] R. Zamir and S. Shamai. Nested linear/lattice codes for Wyner-Ziv encoding. In *Information Theory Workshop (ITW '98)*, June 1998.

[94] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. In *IEEE Workshop on Statistical Signal Processing (SSP '03)*, September 2003.

# Appendix A

# Computation of the Rate-Distortion Bounds

In this Appendix, we provide a more detailed derivation of the different R-D bounds presented in Chapter 4. The piecewise polynomial model that we use is shown in Figure A.1. It corresponds to $L$ separated polynomial pieces of maximum degree $Q$. The $i^{th}$ piece is represented with the polynomial $p_i(t)$ and is defined over the support $I_i = [t_{2i-1}, t_{2i}]$ of width $S_i$.

## A.1 Independent Encoding[1]

The error relative to each quantized discontinuity can be upper bounded by:

$$e_{t_i}^2 \leq A^2 |t_i - \hat{t}_i| \tag{A.1}$$

where $\hat{t}_i$ is the quantized version of $t_i$ using $R_{t_i}$ bits. The quantization error is at most one half of the quantization step, which leads to the following upper bound

[1]Notice that the derivation proposed in this section is inspired from the work in [55].

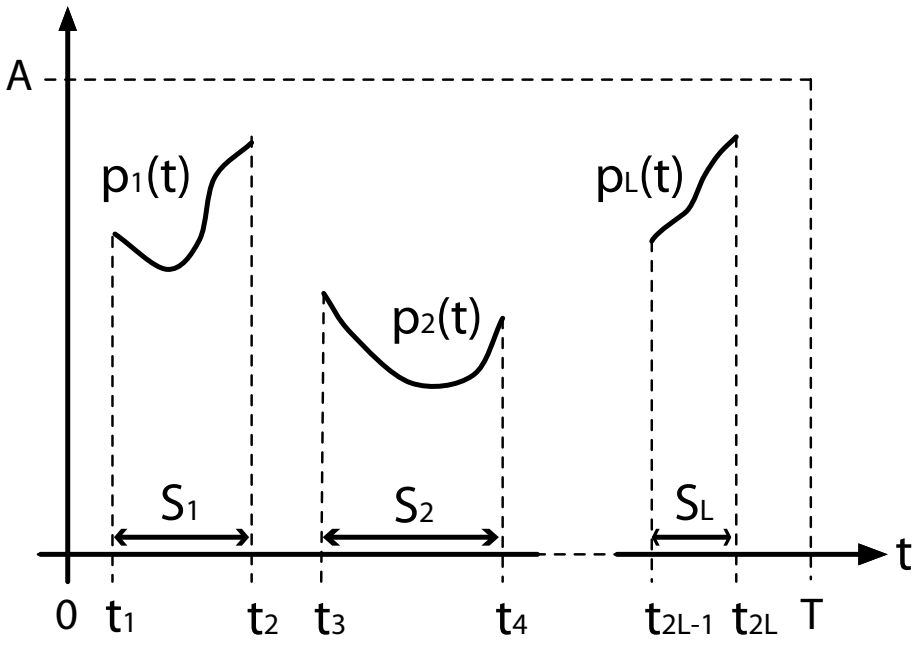


**Figure A.1: Piecewise polynomial signal with $L$ separated pieces of maximum degree $Q$ each. The signal is defined on the suport $[0, T]$ and is bounded in amplitude in $[0, A]$.**

for the distortion associated with the encoding of this discontinuity:

$$D_{t_i}(R_{t_i}) \leq \frac{1}{2} A^2 T 2^{-R_{t_i}} \; . \tag{A.2}$$

Notice that we consider a strict upper bound here (worst case scenario) and therefore do not use the fact that the quantization error is uniformly distributed in $[-T2^{-(R_{t_i}+1)}; T2^{-(R_{t_i}+1)}]$. The different polynomial pieces are encoded using the Legendre expansion. We consider here the $i^{th}$ piece defined over $I_i = [t_{2i-1}, t_{2i}]$: (we deliberately omit the $i$ index for simplicity of notations in the following)

$$p(t) = \sum_{n=0}^{Q} p_n t^n = \sum_{n=0}^{Q} \frac{2n+1}{S} l_n L_I(n;t) \; , \tag{A.3}$$

where $L_I(n;t)$ is the Legendre polynomial of degree $n$ over the support $I$, and $\{l_n\}_{n=0}^{Q}$ are the $Q+1$ Legendre coefficients to be quantized. Notice that due to the properties

of the Legendre expansion, we have:

$$|l_n| \leq \frac{1}{2}AS \tag{A.4}$$

for all $n$. We can thus express the squared error as:

$$\begin{aligned} e^2 &= \sum_{n=0}^{Q} \left(\frac{2n+1}{S}\right)^2 \int_{t_{2i-1}}^{t_{2i}} L_I^2(n;t)dt \\ &= S^{-1} \sum_{n=0}^{Q} (2n+1)(l_n - \hat{l}_n)^2 \,, \end{aligned} \tag{A.5}$$

where $\hat{l}_n$ is the quantized version of $l_n$ using $b_n$ bits (the quantization step is of size $AS2^{-b_n}$ , and $\sum_{n=0}^{Q} b_n = R_p$). The error can therefore be bounded as:

$$e^2 = \frac{1}{4}A^2 S \sum_{n=0}^{Q} (2n+1)2^{-2b_n} \,. \tag{A.6}$$

For the sake of simplicity, we assume that the same number of bits ($b_n = \frac{R_p}{Q+1}$ bits) is allocated to each coefficient, and that the support is maximal ($S = T$). This leads to the following upper bound for the distortion associated with the encoding of this piece:

$$D_p(R_p) \leq \frac{1}{4}A^2 T (Q+1)^2 \, 2^{\frac{-2}{Q+1}R_p} \,. \tag{A.7}$$

The global distortion for the encoded signal can be upper bounded by:

$$D \leq \sum_{i=1}^{L} D_{p_i}(R_{p_i}) + \sum_{i=1}^{2L} D_{t_i}(R_{t_i}) \,. \tag{A.8}$$

The optimal bit allocation for this piecewise polynomial function is impractical to derive because of the dependences on all the polynomial parameters across the whole function. We therefore derive a coarser but more general upper bound by assuming the following simplifications:

- all polynomial pieces are considered as being of degree $Q$.
- the same number of bits $R_p$ is assigned to each polynomial piece.
- all discontinuities are encoded at the same rate $R_t$.

The R-D bound becomes:

$$D(R) \leq \frac{1}{2} A^2 T L (2^{-R_t} + (Q+1)^2 \, 2^{\frac{-2}{Q+1} R_p}) \,, \tag{A.9}$$

where the total rate $R$ corresponds to: $R = L(2R_t + R_p)$. The optimal bit allocation between $R_p$ and $R_t$ can be obtained by computing the derivative of this R-D bound, and is given by:

$$R_p = \frac{Q+1}{Q+5} \frac{R}{L} + G \,, \tag{A.10}$$

$$R_t = \frac{2}{Q+5} \frac{R}{L} - \frac{1}{2} G \,, \tag{A.11}$$

$$\text{where } G = 2(\log(Q+1) + 2) \Big( \frac{Q+1}{Q+5} \Big) \,. \tag{A.12}$$

The global R-D bound is finally given by:

$$D(R) \leq \underbrace{\frac{1}{2} A^2 L T ((Q+1)^2 \, 2^{\frac{-2G}{Q+1}} + 2^{\frac{1}{2}G})}_{c_0} \, 2^{\frac{-2}{(Q+5)L} R} \,. \tag{A.13}$$

## A.2 Distributed Encoding - Scenario A

In this scenario, the total number of bits that needs to be transmitted is given by:

$$R_{tot} = L(R_p + 2R_t + 2R_{t_{SW}}) \,, \tag{A.14}$$

where $R_{t_{SW}} = R_t - \gamma_s$, and $\gamma_s = \lfloor \log_2(\frac{T}{\Delta_{max}}) \rfloor$. We know that distributed encoding using $R_{tot} = L(R_p + 4R_t - 2\gamma_s)$ would lead to an average distortion similar to an

independent encoding of one signal using $R = L(R_p + 2R_t)$ bits. Using the optimal bit allocation computed in the previous section, we obtain:

$$R_{tot} = \frac{Q+9}{Q+5}R - L(2\gamma_s + G)\,, \tag{A.15}$$

$$\Rightarrow\ R = (R_{tot} + L(2\gamma_s + G))\frac{Q+5}{Q+9}\,. \tag{A.16}$$

The global average R-D bound is therefore given by:

$$D_A(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2(2\gamma_s+G)}{Q+9}}}_{c_1} 2^{\frac{-2}{L(Q+9)}R_{tot}}\,. \tag{A.17}$$

## A.3 Distributed Encoding - Scenario B

The total number of bits that needs to be transmitted from the $N \geq L+1$ sensors in this scenario is given by:

$$R_{tot} = L(R_p + 2(L+1)R_t)\,. \tag{A.18}$$

Using the optimal bit allocation, we obtain:

$$R_{tot} = \frac{4L+Q+5}{Q+5}R - L^2G\,, \tag{A.19}$$

$$\Rightarrow\ R = (R_{tot} + L^2G)\frac{Q+5}{4L+Q+5}\,. \tag{A.20}$$

The global average R-D bound is therefore given by:

$$D_B(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2LG}{4L+Q+5}}}_{c_2} 2^{\frac{-2}{L(4L+Q+5)}R_{tot}}\,. \tag{A.21}$$

## A.4 Distributed Encoding - Scenario C

In this scenario, the total number of bits is given by:

$$R_{tot} = L((L + O_{max} + 1)2R_t + (O_{max} + 1)R_p) \,, \tag{A.22}$$

where $O_{max}$ is the maximum number of occluded views for any given object of the scene. Following the optimal bit allocation, we obtain:

$$R_{tot} = \frac{4L + (O_{max} + 1)(Q + 5)}{Q + 5} R + LGO_{max} \,, \tag{A.23}$$

$$\Rightarrow R = (R_{tot} - LGO_{max}) \frac{Q + 5}{4L + (O_{max} + 1)(Q + 5)} \,. \tag{A.24}$$

The global average R-D bound is therefore given by:

$$D_C(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2O_{max}G}{4L+(O_{max}+1)(Q+5)}}}_{c_3} 2^{\frac{-2}{4L^2+L(Q+5)(O_{max}+1)} R_{tot}} \,. \tag{A.25}$$