

Geometry-Driven Distributed Compression of the Plenoptic Function: Performance Bounds and Constructive Algorithms

Nicolas Gehrig, *Student Member, IEEE*, and Pier Luigi Dragotti, *Member, IEEE*,

TIP EDICS: COD-DSS (Distributed Source Coding)

Abstract

In this paper, we study the sampling and the distributed compression of the data acquired by a camera sensor network. The effective design of these sampling and compression schemes requires, however, the understanding of the structure of the acquired data. To this end, we show that the a-priori knowledge of the configuration of the camera sensor network can lead to an effective estimation of such structure and to the design of effective distributed compression algorithms. For idealized scenarios, we derive the fundamental performance bounds of a camera sensor network and clarify the connection between sampling and distributed compression. We then present a distributed compression algorithm that takes advantage of the structure of the data and that outperforms independent compression algorithms on real multi-view images.

I. INTRODUCTION

The advent of the sensor network technology is having a profound impact on the way in which we sense, process and transport signals of interest. The usual one-to-one communication scenario where a single transmitter encodes and transmit the acquired information to a unique receiver is today well understood, but cannot provide satisfactory answers to the many-to-one or many-to-many scenarios that sensor networks bring on the table. Phenomena of interests that are acquired by sensor networks are inherently multi-dimensional and may exhibit very peculiar structures. For an effective design of a sensor network it is therefore fundamental to understand

The authors are with the Communications and Signal Processing Group, Imperial College London. E-mails: {Nicolas.Gehrig03, P.Dragotti}@imperial.ac.uk. Address: Communications and Signal Processing Group, Electrical and Electronic Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, England; Tel: +44 (0)20 759-46192 ; Fax: +44 (0)20 759-46234.

The material in this paper was presented in part at the IEEE ICIP'04, ICIP'05, DCC'06 and ICIP'07. This work was supported in part by DIF-DTC project number 12.6.2. and EOARD 043061.

such structure in order to devise the best sampling-compression-transmission strategy and to understand the interplay between these stages.

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. These constraints oblige to consider several trade-offs that typically involves acquisition accuracy, computational power, capacity of memory, transmission power, delay and battery life duration. In camera sensor networks (CSN), each sensor is equipped with a digital camera and acquires images of a scene of interest from a certain viewing position. 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 [1], the latest technology trends in cameras and sensor platforms are presented.

In this paper, we focus on camera sensor networks and on the many-to-one communication scenario. Moreover, we concentrate mostly on the interplay between sampling and compression, and assume that the underling multi-access channel has a fixed, known capacity. We thus try to understand the fundamental trade-off between the number of cameras and the compression rate.

Due to the spatial proximity of the different cameras, acquired images can be highly dependent. However, since sensors cannot communicate amongst themselves, compression has to be performed locally at each sensor. The problem of performing separate compression and joint reconstruction of correlated sources is known as distributed source coding and has its theoretical foundation in two papers by Slepian and Wolf [2], and Wyner and Ziv [3]. More recently, Pradhan and Ramchandran have proposed a first constructive distributed coding scheme based on channel coding principles [4]. Practical designs based on advanced channel codes such as Turbo codes [5], [6], [7], [8], [9], [10] and LDPC codes [11], [12], [13], [14], [15], [16], [17] have since been proposed. Several researchers have recently used these approaches with correlated visual information to develop distributed video coding algorithms [18], [19], [20], [21], [22] and distributed multi-view image coding schemes [23], [24], [25], [26], [27], [28], [29], and for a nice recent overview of the topic we refer to [30]. However, while all these distributed coding schemes can closely approach the theoretical performance bounds for different binary correlation models or Gaussian models, they are not necessarily the right codes to handle the correlation structure of the visual information acquired by a multi-camera system. This is mainly due to the fact that the real correlation in multi-view data is governed by the rules of perspective geometry and cannot be well represented by the standard correlation models usually used in channel coding. The main novelty of this paper is therefore to exploit the real multi-view correlation by means of a geometrical coding approach.

In this paper, we show that the correlation in the visual data, which is well modelled by the Plenoptic

Function [31], can be estimated using some limited geometrical information about the scene and the position of the cameras. This limited geometrical information can typically be the distance between the cameras and the minimum and maximum depths (i.e. min and max distance between the cameras and any object of the scene). We then 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. Interestingly, such a strategy that does not use conventional channel codes, can be shown to be optimal in some particular cases and allows for a flexible allocation of the transmission bit-rates amongst the encoders. We also prove that in idealized scenarios, such a strategy leads to an exact ‘bit conservation principle’ [32]. Namely, in the high bit-rate regime, the overall reconstruction fidelity at the receiver does not depend on the number of camera sensors deployed when this number is above the critical sampling (minimum number of node necessary to perform perfect reconstruction in absence of compression). Instead, the overall reconstruction fidelity depends only on the overall number of bits received. This is achieved by combining our distributed compression scheme with some recent results on sampling signals with finite rate of innovation [33], [34], [35]. We finally depart from the idealized cases and propose a practical distributed compression algorithm based on quadtree decomposition for real multi-view images. This algorithm preserves the nice properties of the original scheme when applied to idealized scenes, but is also effective on real multi-view images.

The paper is organized as follows: In the next section, we review the notion of the plenoptic function and introduce the specific visual scenarios we will be studying. We also present a preliminary distributed scheme which is at the foundation of all the subsequent results. Section III analyzes the interplay between sampling and compression in camera sensor networks and provides an exact bit conservation result. We then move from theory to practice and propose a quadtree-based distributed image compression algorithm that exploits the geometrical structure of multi-view images in full. Section V provides simulation results to assess the effectiveness of such a scheme. We finally conclude in Section VI.

II. PRELIMINARIES

In this section we recall the notion of plenoptic function and discuss some multi-camera configurations that will be used in the rest of the paper, we will also present a set of preliminary compression results that will be used to derive fundamental performance bounds in camera sensor networks and to devise a new distributed image compression algorithm.

A. *The Plenoptic Function and Problem Set-up*

The plenoptic function was first introduced by Adelson and Bergen in 1991 [31]. It corresponds to the function representing the intensity and chromaticity of the light observed from every position and direction in

the 3-D space, and can therefore be parameterized as a 7-D 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, θ and ϕ give its orientation, τ is the time and ω corresponds to the frequency considered. The measured parameter I_{PF} is simply the intensity of the light observed under these parameters. The high dimensionality of this function, however, makes it difficult to handle.

If there is no restriction on the position of the cameras (i.e., ‘unstructured’ plenoptic function first introduced in [36]), but we fix the time τ and the frequency ω (i.e., grayscale images or separate RGB channels), we obtain a 5-D representation of the plenoptic function. Moreover, camera positions can be constrained to a plane, a line or a point to further remove one, two or three dimensions respectively.

For example, the case when cameras are on a plane leads to the 4-D lumigraph or lightfield [37], [38] parametrization. This parametrization is obtained by using two parallel planes: the focal plane (or camera plane) and the retinal plane (or image plane). If we now assume that cameras are placed along a straight line, we obtain the *Epipolar Plane Image (EPI)* [39] which is a 3-D plenoptic function. The EPI 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. A 2-D slice of EPI is shown in Figure 1. Notice that, in this set-up, points in the world scene are converted into lines in the plenoptic domain. Alternative 3-D plenoptic function can be obtained, for example, by placing cameras on a circle with cameras oriented toward the outside of the circle. In this case, however, a point in the world scene is not converted into a straight line anymore. Notice that plenoptic functions for circular and spherical configurations have been formulated in [40]. Finally, if we constrain the camera position

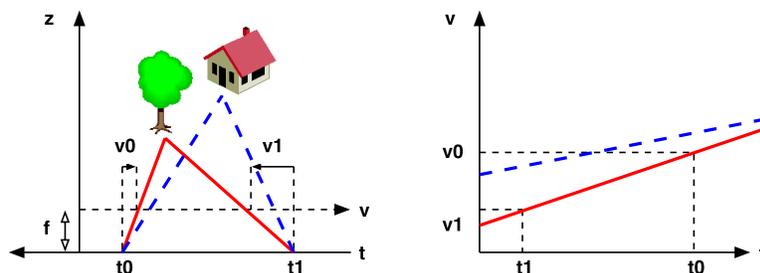


Fig. 1. 2-D plenoptic function of two points of a scene. The t -axis corresponds to the camera position and the v -axis corresponds to the relative positions on the corresponding image. A point of the scene is therefore represented by a line whose slope is directly related to the point’s depth (z -axis). 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.

to a single point, we have a 2-D function which is in fact a common still image.

A camera sensor network is able to acquire a finite number of different views of a scene at any given time and can therefore be seen as a sampling device of the plenoptic function. In this paper, we consider the following

multi-camera scenario: We assume that we have N cameras placed on a horizontal line and that they are all pointing to the same direction (perpendicular to the line of cameras). We assume that the scene is static and that is made of L Lambertian planar polygons that can be horizontally tilted and placed at different depth. Each polygon has an intensity that varies like a polynomial of maximum degree Q . Let α be the distance between two consecutive cameras and assume that the objects have a depth bounded between z_{min} and z_{max} as shown in Figure 2, according to the epipolar geometry principles, which are directly related to the structure of the plenoptic function (see Figure 1), we know that the difference between the positions of a specific point 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. Given α , this disparity Δ 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 encoded irrespective of how complicated the scene is. This fact will be used in the following sections to develop new distributed compression algorithms. Also notice that Δ is bounded also when $z_{max} = \infty$.

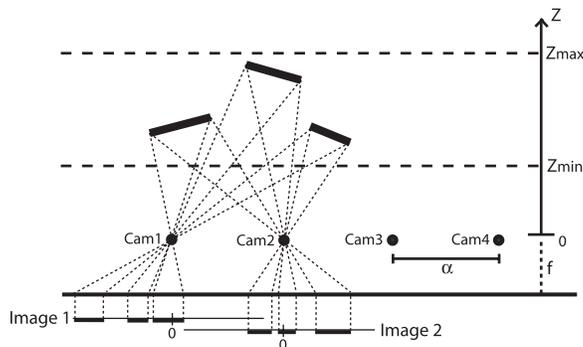


Fig. 2. Our camera sensor network configuration.

It is of interest to point out that the assumption that there is a finite depth of field is not new and was previously used by Chai et al. to develop new schemes for the sampling of the plenoptic function [41]. A similar property is also implicitly used in many mono-video compression algorithms. In that context the assumption is that objects in the sequence move with bounded speed $v \in [v_{min}, v_{max}]$ with $v_{min} \geq 0$ and $v_{max} < \infty$. This allows to restrict the size of the window in which the motion search is performed.

Finally, we would also like to highlight the fact that cameras do not necessarily need to be on a straight line. Alternatively they could be placed on a circle or a plane without changing the main fact that the disparity Δ is bounded. Since these different configurations all share the same property that disparity is bounded and given that this is the only key a-priori information that is used in our distributed compression schemes, we will just consider the linear configuration for the sake of clarity, extension to other structured configurations being straightforward.

B. A Distributed Coding Scheme for Scenes made of Flat Polygons

To illustrate the potential advantage of exploiting any a-priori information about the camera network configuration, we now present a simple lossless distributed compression algorithm for the set-up presented in Figure 2. The proposed algorithm is optimal in some particular situations, namely when the set of disparities and the discontinuity locations are uniformly distributed.

Let us assume that the scene is made of simple objects such as uniformly coloured polygons and that there are only two cameras. Denote with X and Y the horizontal discrete positions (in unit pixel precision) of a specific object or a feature point such as a corner point on the images obtained from the two cameras. Assume that the images are square images with 2^{2R} pixels each (i.e., 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 at least $H(X) + H(Y)$ bits, where $H(\cdot)$ is the entropy of a discrete source. However, by 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 exact values of X and Y is related to the mutual information $I(X, Y)$ and is by definition available at both sources. 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 \tilde{Y} be defined as $\tilde{Y} = Y - \lceil \frac{\alpha f}{z_{max}} \rceil$. This implies that the difference $(\tilde{Y} - X)$ is contained in $\{0, 1, \dots, \delta\}$, where $\delta = \lceil \alpha f (\frac{1}{z_{min}} - \frac{1}{z_{max}}) \rceil$. Looking at the binary representations of X and \tilde{Y} , we realize 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 \tilde{Y}_1 correspond to the last R_{min} bits of X and \tilde{Y} respectively. Let $X_2 = (X \gg R_{min})$ and $\tilde{Y}_2 = (\tilde{Y} \gg R_{min})$, where the “ \gg ” operator corresponds to a binary shift to the right. We thus have that $\tilde{Y}_2 = X_2$ if $\tilde{Y}_1 \geq X_1$ and that $\tilde{Y}_2 = X_2 + 1$ if $\tilde{Y}_1 < X_1$. As indicated in Figure 3, our coding strategy consists in sending X_1 and \tilde{Y}_1 from the sources X and Y respectively and then, sending only complementary subsets of bits for X_2 and \tilde{Y}_2 . At the receiver, X_1 and \tilde{Y}_1 are then compared to determine

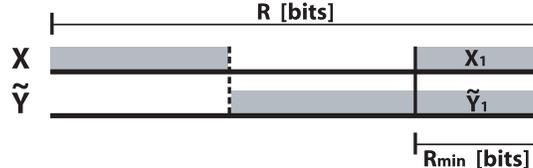


Fig. 3. Binary representation of the two correlated sources. The last R_{min} bits are sent from the two sources but only complementary subsets of the first $(R - R_{min})$ bits are necessary at the receiver for a perfect reconstruction of X and Y .

whether $\tilde{Y}_2 = X_2$ or $\tilde{Y}_2 = X_2 + 1$. Knowing this relation and their partial binary representations, the decoder

can now perfectly recover the values of X and \tilde{Y} .

Assume that z_{min} and z_{max} are such that $(\delta + 1)$ is a power of 2. If the disparity $(\tilde{Y} - X)$ is uniformly distributed, we have that $H(\tilde{Y} - X) = H(X|Y) = H(Y|X) = R_{min}$. Let $S(X_2)$ be a subset of the $R - R_{min}$ bits of X_2 and let $\bar{S}(\tilde{Y}_2)$ corresponds to the complementary subset of \tilde{Y}_2 . If we assume that X is uniformly distributed in $\{0, 1, \dots, 2^R - 1\}$, we have that $H(S(X_2)) + H(\bar{S}(\tilde{Y}_2)) = H(S(X_2), \bar{S}(\tilde{Y}_2)) = I(X, Y)$. The total rate necessary for our scheme is $I(X, Y) + 2R_{min} = H(X, Y)$. In this case, therefore, this scheme is optimal since the total rate is equal to $H(X, Y)$. Moreover, by changing the repartition of bits between X_2 and \tilde{Y}_2 , any point in the Slepian-Wolf rate region can be achieved. For example, the most asymmetric scenario is when the entire X is transmitted from the first encoder and only \tilde{Y}_1 is transmitted by the second encoder. Instead we have a symmetric repartition when X_2 and \tilde{Y}_2 carry the same number of bits as depicted in Figure 3.

III. FUNDAMENTAL TRADE-OFFS IN CAMERA SENSOR NETWORKS

When dealing with sensor networks, one of the fundamental issue is to understand the interplay between sampling and distributed compression. Namely, if we assume that sensors do not communicate among themselves but are only allowed to communicate to the base station through a multi-access channel of fixed capacity, then the fundamental issue is to understand whether it is better to have more sensors but less bandwidth per sensors or vice-versa less sensors, but more bandwidth per sensor.

On the one hand, deploying too few sensors is equivalent to undersample the phenomenon in space and would lead to highly aliased reconstructions. An excessive number of sensors, on the other hand, would use the communication resources inefficiently by transmitting correlated measurements to the receiver and it is not clear whether this inefficiency can always be addressed by means of distributed source coding techniques.

Indeed, preliminary works on the topic have presented pessimistic results regarding the scalability of sensor networks [42], [43]. More optimistic results were more recently proposed in [44], [45], [32]. In those papers, it was shown that, for a given distortion, an upper-bound (independent of N) on the total information rate can be given. The approach in [32], however, requires some communication between neighbouring sensors.

We now aim to understand this fundamental trade-off for the specific case of camera sensor networks. We consider the camera sensor network set-up proposed in Figure 2, and assume that z_{max} can be equal to infinity. Since the world scene is made of L Lambertian planar polygons with polygonal intensity, the perspective projection observed at each camera using the idealized pinhole camera model is therefore given by a 2-D piecewise polynomial function. In practice, however, we only observe a blurred and sampled version of such a projection. The blurring is due to the lenses of the cameras and is normally modelled as a space-invariant filtering with a Gaussian or B-spline filter, the sampling is due to the CCD array. 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. 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.

The N cameras then have to communicate their acquired and processed data to a central station through a multi-access channel with fixed capacity $C = R_{tot}$. We now want to address the two following questions: a) Is there a sampling result that guarantees that perfect reconstruction of this type of 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 the number of cameras going to influence the reconstruction fidelity at the decoder?

A. 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)*. This is because such signals are completely specified by a finite number of parameters (i.e., the discontinuities locations and the coefficients of each polynomial). Recently, new sampling methods for these classes of non-bandlimited signals have been proposed [33], [34]. 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 such as B-splines. Extensions of these sampling approaches to 2-D signals with FRI have been proposed recently [46], [35]. These sampling schemes operate as follows: The acquisition of the signal is linear, namely, the continuous signal is filtered with one of the aforementioned kernels and then sampled. The reconstruction is non-linear and is based on techniques developed in spectral estimation. The fundamental point is that since the original signal is by hypothesis completely specified by a finite number of parameters, exact reconstruction is possible because it is possible to retrieve these parameters from the measured samples.

If we assume that the blurring due to lenses can be modelled as a filtering with one of the above kernels, it is then possible to retrieve, at each sensor, the original piecewise polynomial projection from the observed blurred and sampled version using the above mentioned sampling schemes.

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 views¹. Once this

¹The matching problem can be solved exactly using a finite number of images as shown in [47]. This is due to the fact that the perspective projections in a pinhole camera model are equivalent to the Radon projections in a Radon transform set-up. When the scene is made of polygons, the back-projection problem can be solved exactly using a finite number of projections. We refer to [47] and references therein for further details.

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 scenarios leading to three different sampling results:

Scenario 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).

Scenario 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. We can then back-project the $L + 1$ left extremities of the pieces and retrieve the L real locations of these extremities [47]. 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).

Scenario C: 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 only be either fully visible or fully occluded in any given view.) In this case, 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. We now 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.

B. A Bit-Conservation Result

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 retrieved parameters of the original view are then scalar quantized according to some target distortion for each view. Finally, each quantized parameter is Slepian-Wolf (S-W) encoded using the lossless distributed compression algorithm of Section II-B. More precisely, 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 need to be transmitted only once. The discontinuities can be S-W encoded as proposed in II-B.

In the following, we first describe the rate-distortion behaviour of an encoder that compresses the piecewise polynomial signals independently. Then, we analyze the performance of our distributed compression scheme for the three scenarios highlighted before.

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 by [48]:

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

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:

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

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

Assume now that the N views that have to be transmitted to the receiver are encoded independently. In this case, for an average distortion D over all the reconstructed views at the receiver, the total amount of data to be transmitted is of the order $R_{tot} = NR$ leading to the following global $D(R)$ behaviour:

$$D(R_{tot}) \leq c_0 2^{\frac{-2}{NL(Q+5)} R_{tot}}. \quad (3)$$

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 α 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: $\Delta \in [0; \frac{\alpha f}{z_{min}}]$.

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 $\{\Delta_i\}_{i=1}^{2L}$.

Scenario 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 visual scene.

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.² 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 as shown in Sec. II-B. 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, since the most significant bits can be retrieved from the first view as shown in II-B. Here, $\Delta_{max} = \frac{\alpha f}{z_{min}}$.

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 .

Assume now that we want to transmit information of the scene from more than two encoders. 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 as follows: 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 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. 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

²Notice 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.

the N cameras.³

The number of sensors used for the transmission has therefore 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 VII-B):

$$D_A(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2(2\gamma_s+G)}{Q+9}}}_{c_1} 2^{\frac{-2}{L(Q+9)}R_{tot}}. \quad (4)$$

We can thus observe that Eq.(4) does not depend on the number of sensors N .

Scenario 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: Each new sensor introduced in the system takes the responsibility of transmitting partial information about the polynomial pieces (therefore reducing the communication task of some other sensors) or replaces 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 VII-C):

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

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.

Scenario 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 and the distortion of any reconstructed view given the total bit budget R_{tot} can be shown to behave as (see Appendix VII-D):

$$D_C(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2O_{max}G}{4L+(O_{max}+1)(Q+5)}}}_{c_3} 2^{\frac{-2}{4L^2+L(Q+5)(O_{max}+1)}R_{tot}}. \quad (6)$$

Table I summarizes the different R-D bounds derived in this section (see Appendix VII for more details).

³Notice that in order to perform a symmetric encoding over the N cameras, we need to guarantee a high bit-rate regime where $\frac{R_{tot}}{N} \geq 2LR_{t_{SW}}$.

TABLE I
SUMMARY OF THE DIFFERENT R-D BOUNDS FOR THE ENCODING OF N VIEWS.

Encoding mode	average R-D upper bounds
Indep. Encoding	$D(R_{tot}) \leq c_0 2^{\frac{-2}{NL(Q+5)} R_{tot}}$
DSC - Scenario A	$D_A(R_{tot}) \leq c_1 2^{\frac{-2}{L(Q+9)} R_{tot}}$
DSC - Scenario B	$D_B(R_{tot}) \leq c_2 2^{\frac{-2}{L(4L+Q+5)} R_{tot}}$
DSC - Scenario C	$D_C(R_{tot}) \leq c_3 2^{\frac{-2}{4L^2+L(Q+5)(O_{max}+1)} R_{tot}}$

C. Simulation Results

In order to demonstrate the validity of the above R-D analysis, 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 (32×32 pixels) of the original scene as illustrated in Figure 4 (top). For each view, knowing that the original scene belongs to a certain class of signals with finite rate of innovation, the sampling results of Section III-A 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 (bottom)). Here, the rate-distortion behaviour can be shown to satisfy the expression given in equation 4 as shown on Figure 4 (right) (Notice that in the 2D case, L corresponds to the number of vertices divided by 2).

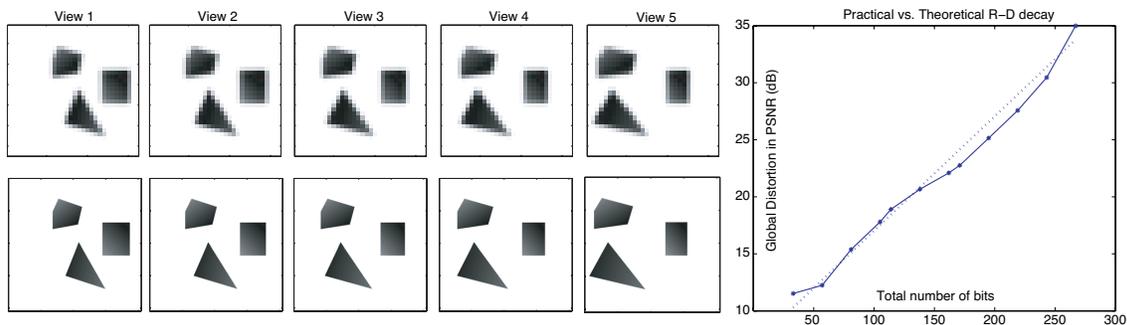


Fig. 4. (Top) Observations at the sensors. Each sensor observes a blurred and undersampled version of the perspective projection of the scene. (Bottom) High resolution version of the 5 views reconstructed at each encoder from these samples using the FRI reconstruction method. (Right) Practical (solid line) vs. Theoretical (dotted line) rate-distortion behaviour.

The original views are represented with the following precision: 22 bits for each vertex (view of 2048×2048 pixels) and 9 bits for each polynomial coefficient. One view is therefore exactly represented using $12 \times 22 + 9 \times 9 = 345$ bits. The parameters α , 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 III-B2,

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 II highlights our bit-conservation result for different bit budgets R_{tot} . Indeed, it shows that our approach suffers no rate loss when we increase the number of sensors used for transmission. Notice that the PSNR of a reconstruction is given as the average value of the PSNRs of all reconstructed views.

TABLE II
AN EXACT BIT-CONSERVATION RESULT: SIMULATION RESULTS WITH THE 5 VIEWS PRESENTED IN FIGURE 4.

R_1 (bits)	R_2 (bits)	R_3 (bits)	R_4 (bits)	R_5 (bits)	R_{tot} (bits)	PSNR (dB)
345	-	-	-	120	465	∞
276	-	-	-	108	384	44
128	-	128	-	128	384	44
108	84	-	84	108	384	44
84	72	72	72	84	384	44
171	-	-	-	60	231	29
48	48	39	48	48	231	29

IV. TREE-STRUCTURED APPROACHES FOR COMPRESSION OF MULTI-VIEW IMAGES

Equipped with the results of the previous sections, we now aim to develop a practical distributed compression scheme that, for the idealized scenarios, achieves the arbitrary partition of rate amongst encoders and the bit-conservation results of the previous section, but that can also operate effectively on real multi-view images.

A. The prune-join decomposition algorithm

In [49], 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 1-D (using binary trees) or for 2-D (using quadtrees) signals.

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 and J_{max} is the maximum number of binary tree decomposition levels. 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 algorithm then generates R-D curves for each node of the binary tree by quantizing uniformly 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 λ 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 λ . In order to further improve the encoding performance, 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 λ must be updated and the pruning and joining procedure must be repeated until it reaches the global target rate or distortion.

We give a sketch of this compression algorithm for 1-D signals in the table ‘Algorithm 1’ and encourage the reader to refer to the original work [49] for more details.

1) *A distributed coding strategy for 1-D piecewise polynomial functions:* Let $f_1(t)$ and $f_2(t)$ be two 1-D piecewise polynomial views obtained from two different cameras as defined in Section III-B2. Assume that these two signals are independently encoded using Algorithm 1 for a given distortion target. The total information necessary to describe each of them can be divided into 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.

Algorithm 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 λ and go back to point 4).
-

Figure 5 shows 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. This similarities are exploited in our distributed compression algorithm 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_Δ 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_Δ corresponds to the number of nodes in the pruned tree with a depth smaller than J_Δ .

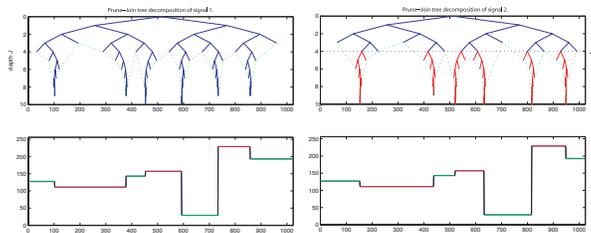


Fig. 5. Prune-Join binary tree decomposition of two piecewise constant signals satisfying our correlation model.

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.

This encoding strategy is asymmetric. In order to allow for an arbitrary bit-rate allocation between the two encoders, one can devise the following alternative strategy:

- Send all the subtrees having a root node at level J_Δ 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.

Assume now that we want to transmit some information from more than two cameras. We have previously seen that if we assume that there are no occlusions in different views, the knowledge of the discontinuity locations from only two cameras is sufficient to reconstruct any view in between. It is easy to realize 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. Moreover, since the polynomial pieces are similar for each view, they can be transmitted from any camera, all this without impairing reconstruction quality. Numerical experiments in Section V-A will confirm that this strategy leads to a bit conservation result for simple scenes.

B. Extension to 2-D using Quadtree Decompositions

The prune-join binary tree decomposition used in the 1-D case has an intuitive extension in 2-D. In this case, the binary tree is replaced with a quadtree and the polynomial model is replaced with a 2-D geometrical model⁴. Algorithm 2 provides a sketch of our implementation of the quadtree compression approach proposed in [49]. Figure 6(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 6(c)) has a higher PSNR (about 1dB) than the one obtained using a Jpeg2000 encoder (we use the java implementation of the Jpeg2000, reference software available at: <http://jj2000.epfl.ch>).

In much the same way as in the 1-D scenario, our distributed 2-D coding approach consists in decomposing each view using the quadtree approach presented in Algorithm 2, and then transmitting only partial information from each view. The total information necessary to describe each view can be divided into three 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 leaf. Finally, R_{Leaves} is the total

⁴The geometrical model used is sketched in Fig. 6(a). This model is optimal when contours in images satisfy some regularity constraints (i.e., contours should be \mathbb{C}_2 functions [49]). In more realistic settings this model provides the best R-D performance at low rates.

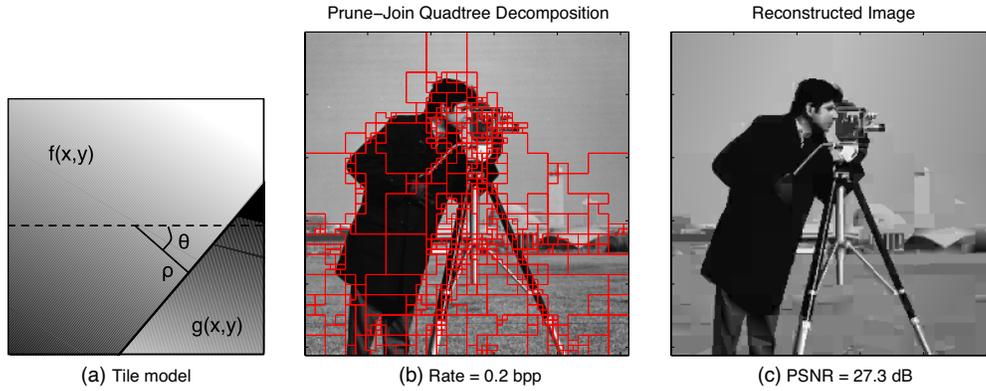


Fig. 6. Quadtree Decomposition. (a) Our geometrical model consists of two 2D linear regions separated by a 1D linear boundary. (b) Prune-Join quadtree decomposition of cameraman with a target bit-rate of 0.2 bpp. (c) The PSNR of the reconstructed image is about 1dB better than what we obtain with a Jpeg2000 encoder.

Algorithm 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 regions of linear intensities separated by a 1D linear boundary (see Figure 6(a)). For both regions, the intensity along the horizontal and vertical axis varies as a linear 2D polynomial.
 - 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,right).
 - 6: Search for the desired R-D operating slope (update λ and go back to 4).
-

number of bits necessary to code the geometrical information of the leaves (2-D polynomials, 1-D boundaries and model side information).

The distributed compression strategy for two views can therefore be described as follows (asymmetric case):

- Send the full description of the first view from the first encoder.
- Send only the subtrees of the quadtree structure of the second view with 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 1-D 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 as described in the 1-D case. In order to give the correct intuition about this coding approach, we propose a simple example in Section V-B, where the scene consists of a simple rectangle of constant intensity.

C. 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. 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 satisfies 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. 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 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. The amount of most significant bits to be transmitted has been determined empirically after running several simulations on various datasets.

During the reconstruction, each region of the quadtree of the second view is matched with its corresponding region in the first view. This matching is done by scanning all the possible positions on the first view according to the epipolar constraints and selecting the one that is closest to the characteristics obtained from the second

view (i.e., 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. Notice also that since the distributed encoding of disparity here is slightly different from the one discussed before, the bit conservation results of the previous section does not hold exactly in the case of real multi-view images.

The main difficulty with the proposed geometry compensation approach is that it tries to match two regions without having access to all the pixels information. A simple distortion metric such as the mean square error (MSE) can therefore not be used to evaluate the best match. For each area corresponding to a tile (or a group of joint tiles) on the second view, the proposed approach first generates the set of all possible matching regions on the first view by shifting the left and right extremities of the region by all the values between the minimum and maximum disparities. Then, in order to select the best candidate, all these potential regions of the first view are approximated using the geometrical model parameters provided by the second view. These constrained approximations are then evaluated by computing their mean square errors with the original data of the first view and the candidate presenting the minimum distortion is selected as the best match. Finally, the missing polynomials for the considered area of the second view are simply estimated from this retrieved area of the first view.

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

V. SIMULATION RESULTS

A. 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 fact that also the proposed tree-structured compression algorithm allows for an arbitrary bit-rate allocation and as presented in Section IV-A1.

In Table III, we highlight the bit conservation result, by applying our compression approach to the three views shown in Figure 7. 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 compression 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 (an independent encoder would need 438bits). A similar global rate-distortion behaviour holds when part of the information is transmitted from the central view.⁵ This means that, if we assume that the sensors transmit their compressed data to the central decoder using a multi-access channel, the fidelity of the reconstructed views

⁵The small variations in the SNR values are only due to different quantization errors.

depends only on the global capacity of this channel and not on the number of sensors used. The last line of Table III shows a result obtained with another assumption on z_{min} . Here, the new z_{min} is chosen such that the maximum disparity is doubled and we observe that the algorithm only requires 4 more bits to reach the same global distortion.

TABLE III

DISTRIBUTED COMPRESSION USING A 1-D PRUNE-JOIN ALGORITHM. SIMULATION RESULTS WITH THREE 1-D SIGNALS.

Coding Strategy	R_1 (bits)	R_2 (bits)	R_3 (bits)	R_{tot} (bits)	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
SW - 3 views	86	67	86	239	26.13

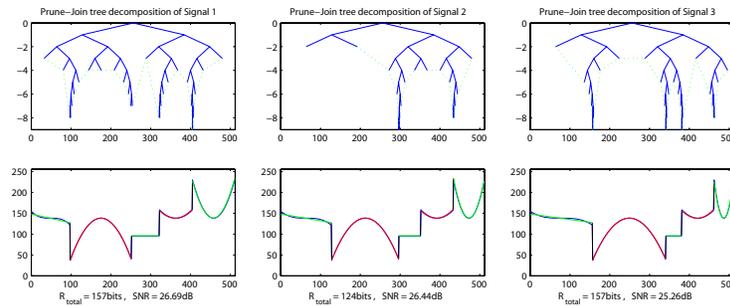


Fig. 7. Join-Prune tree decomposition of three piecewise polynomial signals for a given distortion target (26 dB).

B. A simple example in an ideal scenario

We consider here a simple example to highlight our quadtree based distributed coding technique. Figure 8 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 IV, 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 IV presents an example of bit allocation between the three encoders, where 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). The last column of Table IV shows the joint entropy of the source therefore proving that the approach achieves the Slepian-Wolf bounds.

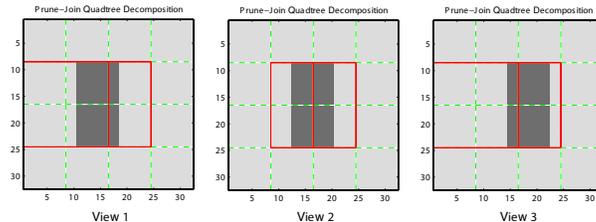


Fig. 8. Three 32×32 views of a very simple synthetic scene consisting of one rectangle at a certain depth. The quadtree decomposition uses a total of 114 bits to encode each view (21 bits for the pruned quadtree, 42 bits for the joining information and 51 bits for the geometrical representations).

TABLE IV

NUMERICAL RESULTS FOR THE LOSSLESS ENCODING OF THE SIMPLE VIEWS SHOWN IN FIGURE 8

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

C. Simulation results on real 2-D multi-view images

We now present simulation results obtained on the *lobby* multi-view sequence from Shum et al. [50]. Figure 9 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 10 and 14(a) 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 14(a) shows that our approach outperforms an independent encoding of the six views for all the range of considered bit-rates. In Figures 11, the difference in the reconstruction quality between the independent and distributed approach is highlighted. The images correspond to the 5th view of the lobby sequence. The error of the distributed quad-tree-based scheme is typically due to the noise, the non-lambertianity of the scene, or the presence of non-flat surfaces.

Figures 12 and 13 show other simulation results obtained on different set of multi-view images. Again, the distributed approach outperforms the independent compression algorithm as shown in Figure 14(b). In Figure 14, we also show the results obtained with an independent JPEG2000 coder as well as an existing distributed video coder named DISCOVER [22] which uses standard channel codes to perform distributed compression. In order to use this video coder on our set of multi-view images in a comparable approach, we consider the 6 views as 6 video frames where the first and the last frames are intra-coded and the four intermediate frames are Wyner-Ziv coded. The resulting R-D curves show that our approach performs clearly better than the DISCOVER approach at low bit-rate despite the fact that DISCOVER uses a feedback channel to perform rate control. Such channel does not exist in truly distributed coding scenarios and is not used in our scheme.

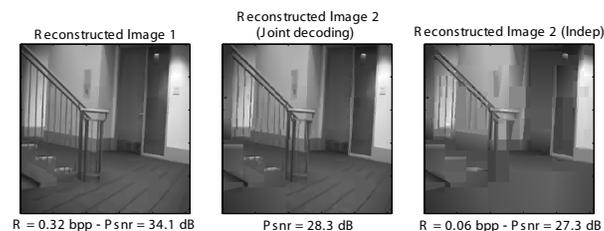


Fig. 9. Distributed stereo encoding of two views. View 1 (left) is encoded at 0.32 bpp and fully transmitted. View 2 (right) is encoded at a much lower bit-rate and some of its polynomial coefficients are discarded. Joint decoding of view 2 (center) shows an improvement of the reconstruction quality of about 1dB compared to an independent encoding.

VI. CONCLUSIONS

In this paper the problem of sampling and distributed compression of the data acquired by a camera sensor network has been considered. First, the structure of the multi-view data has been analyzed and it was shown that it is possible to devise distributed compression strategies that exploit any a-priori knowledge of the configuration of the multi-camera system. Moreover, for idealized scenario, we demonstrated that the overall reconstruction

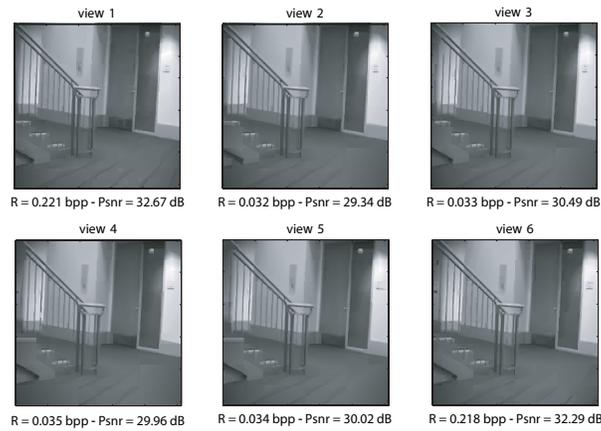


Fig. 10. The six images correspond to a distributed encoding with an average bit-rate of 0.08 bpp. Images 1 and 6 are encoded independently while images 2 to 5 are encoded using our distributed compression approach.

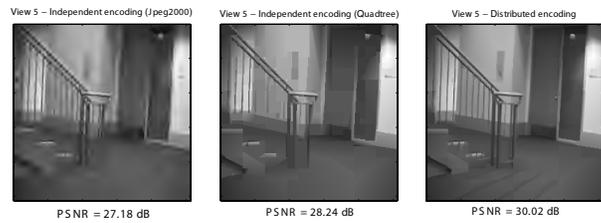


Fig. 11. Reconstruction of the 5th view. Independent encoding at 0.08 bpp with Jpeg2000 (left). Independent encoding at 0.08 bpp with the prune-join quadtree coder (center). Our distributed encoding approach with an average of 0.08 bpp (right).

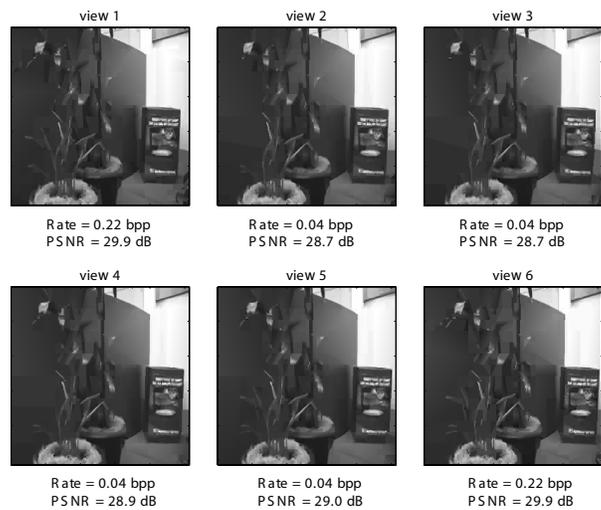


Fig. 12. Distributed encoding of the six views with an average bit-rate of 0.1 bpp. Images 1 and 6 are encoded independently while images 2 to 5 are encoded using our distributed compression approach.

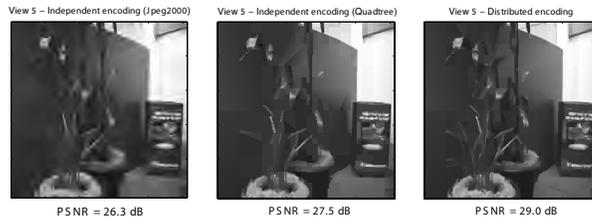


Fig. 13. Reconstruction of the 5th view. Independent encoding at 0.1 bpp with Jpeg2000 (left). Independent encoding at 0.1 bpp with the prune-join quadtree coder (center). Our distributed encoding approach with an average of 0.1 bpp (right).

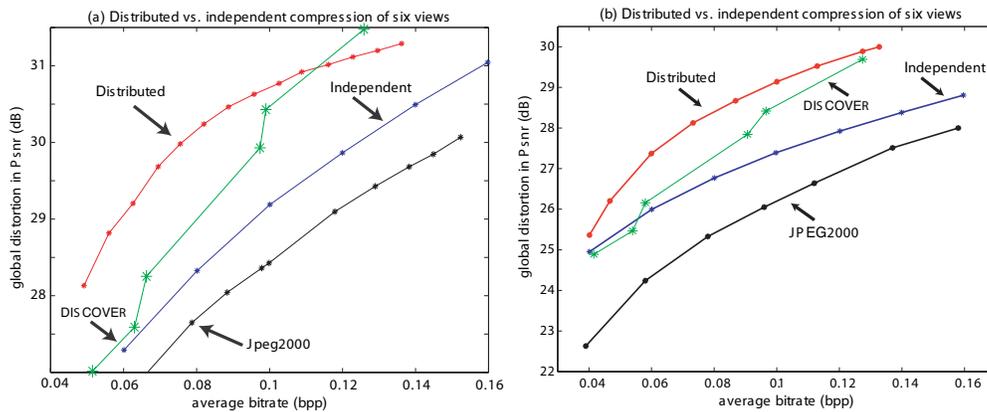


Fig. 14. Distributed vs. independent encoding of six views with our quadtree-based approach. Results obtained with an existing distributed video coder (DISCOVER [22]) and a Jpeg2000 encoder are also shown. (a) Results for the scene shown in Figure 10. (b) Results for the scene shown in Figure 12.

fidelity does not depend on the number of cameras involved but only on the total number of bits transmitted to the receiver, this leads to an exact bit-conservation result.

We have then considered the problem of compressing real multi-view images and extended a quadtree-based compression scheme to the case of distributed image compression. The algorithm takes advantage of the structure of the visual data and of the configuration of the camera system and perform well on real multi-view images. We also showed that the proposed scheme achieves the ideal performance when dealing with synthetic images.

In future research, we will consider different camera configurations and the case of multi-view video sequences.

VII. APPENDIX - COMPUTATION OF THE RATE-DISTORTION BOUNDS

In this Appendix, we provide a more detailed derivation of the different R-D bounds presented in Section III. The piecewise polynomial model that we use 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. Independent Encoding⁶

The error relative to each quantized discontinuity can be upper bounded by: $e_{t_i}^2 \leq A^2 |t_i - \hat{t}_i|$, 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 for the distortion associated with the encoding of this discontinuity:

$$D_{t_i}(R_{t_i}) \leq \frac{1}{2} A^2 T 2^{-R_{t_i}}. \quad (7)$$

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)

$$p(t) = \sum_{n=0}^Q p_n t^n = \sum_{n=0}^Q \frac{2n+1}{S} l_n L_I(n; t), \quad (8)$$

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: $|l_n| \leq AS/2$ for all n . We can thus express the squared error as:

$$e^2 = \sum_{n=0}^Q \left(\frac{2n+1}{S} \right)^2 \int_{t_{2i-1}}^{t_{2i}} L_I^2(n; t) dt = S^{-1} \sum_{n=0}^Q (2n+1) (l_n - \hat{l}_n)^2, \quad (9)$$

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:

$$e^2 = \frac{1}{4} A^2 S \sum_{n=0}^Q (2n+1) 2^{-2b_n}. \quad (10)$$

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:

$$D_p(R_p) \leq \frac{1}{4} A^2 T (Q+1)^2 2^{\frac{-2}{Q+1} R_p}. \quad (11)$$

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

$$D \leq \sum_{i=1}^L D_{p_i}(R_{p_i}) + \sum_{i=1}^{2L} D_{t_i}(R_{t_i}). \quad (12)$$

The optimal bit allocation for this piecewise polynomial function is impractical to derive because of the dependencies on all the polynomial parameters across the whole function. We therefore derive a coarser but

⁶Notice that the derivation proposed in this section is mostly based on the work in [48].

more general upper bound by assuming the following simplifications: a) All polynomial pieces are considered as being of degree Q . b) The same number of bits R_p is assigned to each polynomial piece. c) All discontinuities are encoded at the same rate R_t . The R-D bound becomes:

$$D(R) \leq \frac{1}{2} A^2 T L (2^{-R_t} + (Q+1)^2 2^{\frac{-2}{Q+1} R_p}), \quad (13)$$

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:

$$R_p = \frac{Q+1}{Q+5} \frac{R}{L} + G, \quad R_t = \frac{2}{Q+5} \frac{R}{L} - \frac{1}{2} G,$$

where $G = 2(\log(Q+1) + 2) \left(\frac{Q+1}{Q+5} \right)$. The global R-D bound is finally given by:

$$D(R) \leq \frac{1}{2} \underbrace{A^2 L T ((Q+1)^2 2^{\frac{-2G}{Q+1}} + 2^{\frac{1}{2}G})}_{c_0} 2^{\frac{-2}{(Q+5)L} R}. \quad (14)$$

B. Distributed Encoding - Scenario A

In this scenario, the total number of bits that needs to be transmitted is given by: $R_{tot} = L(R_p + 2R_t + 2R_{t_{sw}})$, 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:

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

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

$$D_A(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2(2\gamma_s+G)}{Q+9}}}_{c_1} 2^{\frac{-2}{L(Q+9)} R_{tot}}. \quad (15)$$

C. 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: $R_{tot} = L(R_p + 2(L+1)R_t)$. Using the optimal bit allocation, we obtain:

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

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

$$D_B(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2LG}{4L+Q+5}}}_{c_2} 2^{\frac{-2}{L(4L+Q+5)} R_{tot}}. \quad (16)$$

D. Distributed Encoding - Scenario C

In this scenario, the total number of bits is given by: $R_{tot} = L((L + O_{max} + 1)2R_t + (O_{max} + 1)R_p)$, where O_{max} is the maximum number of occluded views for any given object of the scene. Following the optimal bit allocation, we obtain:

$$R_{tot} = \frac{4L + (O_{max} + 1)(Q + 5)}{Q + 5}R + LGO_{max}, \quad (17)$$

$$\Rightarrow R = (R_{tot} - LGO_{max})\frac{Q + 5}{4L + (O_{max} + 1)(Q + 5)}. \quad (18)$$

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

$$D_C(R_{tot}) \leq \underbrace{c_0 2^{\frac{-2O_{max}G}{4L + (O_{max} + 1)(Q + 5)}}}_{c_3} 2^{\frac{-2}{4L^2 + L(Q + 5)(O_{max} + 1)}} R_{tot}. \quad (19)$$

REFERENCES

- [1] 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)*, June 2005, pp. 141–146.
- [2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.
- [3] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, January 1976.
- [4] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Data Compression Conference*, 1999, pp. 158–167.
- [5] J. Garcia-Frias, "Compression of correlated binary sources using Turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, October 2001.
- [6] A. Aaron and B. Girod, "Compression with side information using Turbo codes," in *IEEE International Conference on Data Compression*, April 2002, pp. 252–261.
- [7] A. Liveris, Z. Xiong, and C. Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," in *Data Compression Conference (DCC 2003)*, March 2003, pp. 193–202.
- [8] J. Li, Z. Tu, and R. Blum, "Slepian-Wolf coding for nonuniform sources using turbo codes," in *Data Compression Conference (DCC 2004)*, March 2004, pp. 312–321.
- [9] 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, p. 330.
- [10] 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)*, December 2005, pp. 352–357.
- [11] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes," *IEEE Comm. Letters*, vol. 6, no. 10, pp. 440–442, October 2002.
- [12] 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*, October 2002.
- [13] A. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades, "Slepian-Wolf coding of three binary sources using ldpc codes," in *Intl. Symp. Turbo Codes and Related Topics*, September 2003.

- [14] 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.
- [15] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," in *Data Compression Conference (DCC 2004)*, March 2004.
- [16] 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 2004)*, March 2004.
- [17] M. Sartipi and F. Fekri, "Distributed source coding in wireless sensor networks using LDPC codes: a non-uniform framework," in *Data Compression Conference (DCC 2005)*, March 2005, p. 477.
- [18] R. Puri and K. Ramchandran, "PRISM: A video coding architecture based on distributed compression principles," EECS Department, University of California, Berkeley, Tech. Rep., October 2002, article presented in Proc. Of 40th Allerton Conference on Communication, Control, and Computing, Allerton, IL, Oct. 2002.
- [19] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, January 2005, invited Paper.
- [20] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: An error-resilient compression framework," *IEEE Trans. on Multimedia*, vol. 6, no. 2, pp. 249–258, April 2004.
- [21] Q. Xu, V. Stankovic, and Z. Xiong, "Wyner-Ziv video compression and fountain codes for receiver-driven layered multicast," in *PCS'04*, December 2004.
- [22] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Oualet, "The DISCOVER codec: Architecture, Techniques and Evaluation," in *Picture Coding Symposium (PCS '07)*, November 2007.
- [23] X. Zhu, A. Aaron, and B. Girod, "Distributed compression for large camera arrays," in *Proceedings of the IEEE Workshop on Statistical Signal Processing, SSP-2003*, September 2003.
- [24] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compression of lightfield rendered images using coset codes," in *Asilomar Conf. on Signals and Systems, Special Session on Distributed Coding*, 2003.
- [25] 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)*, September 2004.
- [26] 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, pp. 819–822.
- [27] M. P. Tehrani, T. Fujii, and M. Tanimoto, "Distributed source coding of multiview images," in *SPIE-IST, Electronic Imaging, VCIP*, January 2004, pp. 300–309.
- [28] —, "The adaptive distributed source coding of multi-view images in camera sensor networks," *IEICE Trans. Fundamentals*, vol. E88-A, no. 10, pp. 2835–2843, October 2005.
- [29] 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.
- [30] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, and R. Leonardi, "Distributed Monoview and Multiview Video Coding," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 67–76, September 2007.
- [31] E. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. MIT Press, Cambridge, MA, 1991, pp. 3–20.
- [32] 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.
- [33] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, June 2002.

- [34] P. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1741–1757, May 2007.
- [35] P. Shukla and P. Dragotti, "Sampling schemes for multidimensional signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3670–3686, July 2007.
- [36] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *SIGGRAPH'95*, August 1995, pp. 39–46.
- [37] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *SIGGRAPH, Computer Graphics Proceedings*. ACM Press, 1996, pp. 43–54.
- [38] M. Levoy and P. Hanrahan, "Light field rendering," in *SIGGRAPH, Computer Graphics Proceedings*. ACM Press, 1996, pp. 31–40.
- [39] R. C. Bolles and H. H. Baker, "Epipolar-plane image analysis: A technique for analyzing motion sequences," in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, M. A. Fischler and O. Firschein, Eds. Los Altos, CA.: Kaufmann, 1987, pp. 26–36.
- [40] T. Naemura, T. Takano, M. Kaneko, and H. Harashima, "Ray-based creation of photo-realistic virtual world," in *Int. Conf. on Virtual Reality and Multimedia (VSMM'97)*, September 1997, pp. 59–68.
- [41] 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*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 307–318.
- [42] 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.
- [43] 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, pp. 161–168.
- [44] A. Kashyap, L. Lastras-Montano, C. Xia, and L. Zhen, "Distributed source coding in dense sensor networks," in *Data Compression Conference (DCC 2005)*, March 2005, pp. 13–22.
- [45] D. Neuhoff and 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.
- [46] I. Maravic and M. Vetterli, "Exact sampling results for some classes of parametric non-bandlimited 2-D signals," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 175–189, January 2004.
- [47] A. Chebira, P. L. Dragotti, L. Sbaiz, and M. Vetterli, "Sampling and Interpolation of the Plenoptic Function," in *IEEE International Conference on Image Processing*, September 2003.
- [48] P. Prandoni and M. Vetterli, "Approximation and compression of piecewise smooth functions," *Phil. Trans. R.Soc.Lond.* 1999., vol. 357, no. 1760, pp. 2573–2591, September 1999.
- [49] R. Shukla, P. Dragotti, M. Do, and M. Vetterli, "Rate-distortion optimized tree structured compression algorithms for piecewise polynomial images," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 343–359, March 2005.
- [50] H.-Y. Shum, S. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, November 2003.



Nicolas Gehrig received the PhD degree in March 2008 from the Electrical and Electronic Engineering Department of Imperial College London, where he was a research student in the Communications and Signal Processing group from 2003 to 2007. In March 2003, he graduated from the Swiss Federal Institute of Technology of Lausanne (EPFL), Switzerland, where he obtained the Master degree in Communication Systems. From September to December 2005, he was a visiting scholar at the BASiCS research group, EECS Department, University of California at Berkeley. From September 2002 to May 2003, he worked as a computer vision researcher for Dartfish Ltd, a Swiss company that specializes in the development of video analysis software for sports. From April to October 2001, he interned as a software engineer at the IBM T.J.Watson Research Center in Hawthorne, New York.

Dr Gehrig is now the CEO of Odus Technologies Ltd, a young start-up based in Vevey, Switzerland, that specializes in the research and development of innovative products for dental and medical applications.



Pier Luigi Dragotti is currently a Senior Lecturer (Associate Professor) in the Electrical and Electronic Engineering Department at Imperial College, London. He received the Laurea Degree (summa cum laude) in Electrical Engineering from the University Federico II, Naples, Italy, in 1997; the Master degree in Communications Systems from the Swiss Federal Institute of Technology of Lausanne (EPFL), Switzerland in 1998; and PhD degree from EPFL, Switzerland, in April 2002. In 1996, he was a visiting student at Stanford University, Stanford, CA, and, from July to October 2000, he was a summer researcher in the Mathematics of Communications Department at Bell Labs, Lucent Technologies, Murray Hill, NJ. Before joining Imperial College in November 2002, he was a senior researcher at EPFL for the Swiss National Competence Center in Research on Mobile Information and Communication Systems.

Dr Dragotti is an associate editor of the IEEE Transactions on Image Processing and a member of the IEEE Image and MultiDimensional Signal Processing (IMDSP) Technical Committee. His research interests include: wavelet theory, sampling theory, distributed source coding, image compression and image super-resolution.