

# DISTRIBUTED COMPRESSION OF MULTI-VIEW IMAGES USING A GEOMETRICAL CODING APPROACH

*Nicolas Gehrig and Pier Luigi Dragotti*

Communications and Signal Processing Group, Electrical and Electronic Engineering Department  
Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom  
e-mail: {nicolas.gehrig, p.dragotti}@imperial.ac.uk

## ABSTRACT

In this paper, we propose a distributed compression approach for multi-view images, where each camera efficiently encodes its visual information locally without requiring any collaboration with the other cameras. Such a compression scheme can be necessary for camera sensor networks, where each camera has limited power and communication resources and can only transmit data to a central base station. The correlation in the multi-view data acquired by a dense multi-camera system can be extremely large and should therefore be exploited at each encoder in order to reduce the amount of data transmitted to the receiver. Our distributed source coding approach is based on a quadtree decomposition method and uses some geometrical information about the scene and the position of the cameras to estimate this multi-view correlation. We assume that the different views can be modelled as 2D piecewise polynomial functions with 1D linear boundaries and show how our approach applies in this context. Our simulation results show that our approach outperforms independent encoding of real multi-view images.

*Index Terms*— Multi-view Image Compression, Distributed Source Coding, Camera Sensor Networks.

## 1. INTRODUCTION

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) [1] and lightfield rendering (LFR) [2] 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 redundant 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. This approach 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. Another alternative would be to allow the cameras to communicate amongst themselves to perform the joint compression in a collaborative way before transmitting the compressed data to the central receiver. Nevertheless, these two approaches would clearly use a tremendous amount of communication resources, and might not be feasible in some practical settings. In particular, we consider

the case of camera sensor networks, where each sensor is a self-powered wireless device containing a digital camera and a processing/communication unit. Since inter-sensor communication could be extremely expensive in terms of power consumption in this scenario, we would like to exploit the correlation in the multi-view data at each camera, without requiring any collaboration between the different encoders. Thanks to results obtained by Slepian-Wolf [3] and Wyner-Ziv [4], we know that in many cases, we can theoretically achieve the compression rate of a joint-encoder using separate encoders, assuming that the correlation of the source is known a-priori.

Practical distributed source coding schemes inspired from these theoretical results have been proposed in the last decade, and are mainly based on channel coding principles. In particular, they have more recently led to the development of several distributed video coding approaches, where they allow for a swap of complexity between the encoder and decoder (see [5, 6]).

In [7], we showed how the correlation in the information acquired by a multi-view system can be estimated using simple geometrical constraints on the scene and on the position of the cameras. We then proposed a coding approach that can exploit this correlation in order to perform distributed compression of the different views. Nevertheless, our approach was based on an a-priori knowledge of the different object boundaries at the encoders and was therefore not directly applicable to encode real multi-view images. In [8], we showed how a particular encoder based on tree structured algorithms [9] can be modified to take advantage of our distributed coding approach. In particular, we focused on the 1D case where we showed how scan-lines of different views could be modelled using piecewise polynomial models, and compressed in a distributed way using the a-priori correlation structure.

In this paper, we extend our results by showing how the prune-join binary tree decomposition used in the 1D case has an intuitive extension in the 2D case, where the binary tree is replaced with a quadtree segmentation 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 [9], 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. Notice that several other approaches for distributed compression of multi-view images have been proposed (see [10, 11, 12, 13, 14] for example). The main difference is that these approaches rely on the use of advanced channel coding techniques to exploit the correlation in the multi-view data, whereas our scheme tries to estimate this correlation structure using a fully geometrical approach and does not

---

This work was supported in part by DIF-DTC project number 12.6.2. and EOARD 043061.

require the use of channel codes.

The paper is organized as follows: The next section introduces our camera sensor network set-up and gives a brief review of some of our previous results. In Section 3, we first present the quadtree decomposition algorithm and the correlation model we consider. Then our distributed coding approach is presented focusing on the simple case of a stereo pair of images (two views). Finally, the joint reconstruction method used at the decoder is addressed. Section 4 presents the simulation results and concluding remarks are given in Section 5.

## 2. REVIEW OF OUR PREVIOUS RESULTS

### 2.1. Our linear camera sensor network scenario

Our camera sensor network set-up is illustrated in Figure 1. We assume that  $N$  cameras are placed on a horizontal line (all the cameras are looking perpendicularly to this line) and that all the objects of the scene have a distance to the cameras that is bounded between a minimum and a maximum value ( $z_{min}, z_{max}$ ). Assuming that the distance between two cameras is not larger than a certain distance  $\alpha$ , this scenario ensures that, regardless of the complexity of the scene, any disparity  $\Delta$  will be contained in the range:  $[\frac{\alpha f}{z_{max}}; \frac{\alpha f}{z_{min}}]$  where  $f$  is the (common) focal length of the cameras. Based on this observation, our distributed coding strategy [7] consists in sending only partial information of the positions of the objects from the different encoders, as recalled in Section 2.2.

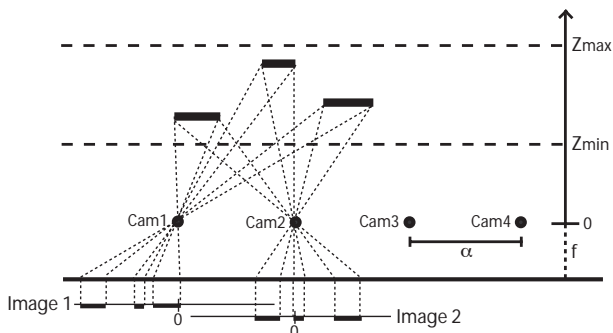


Fig. 1. Our linear camera sensor network set-up

### 2.2. Our distributed compression approach for 1D signals using tree structured algorithms

In [7], we showed how the positions of a specific object on two images obtained from consecutive cameras can be encoded in a distributed way. The method consists in sending subsets of bits from the positions' binary representations such that, knowing the a-priori correlation structure, the decoder can still reconstruct the two positions perfectly. Our approach was shown to allow for a flexible allocation of the bit-rates amongst the encoders and could be made resilient to a fixed number of occlusions.

In [8], we showed that corresponding scan-lines of different views could be represented using a piecewise polynomial model where each discontinuity is shifted from one view to the other according to our correlation model (disparities are bounded in  $\{\Delta_{min}; \Delta_{max}\}$ ). Since we assume that the scene is composed of lambertian planar surfaces and that there is no occlusion, we know that the polynomials are similar in both views.

Piecewise polynomial signals can be efficiently encoded using the prune-join binary tree compression algorithm proposed in [9]. Figure 2 presents a pair of correlated signals (shifted discontinuities) and their binary tree decompositions. Each signal is encoded and represented with a tree structure that models its segmentation and a set of polynomials. Our distributed encoding approach consists in transmitting only the necessary information to allow for a complete reconstruction at the decoder. Each encoder typically transmits partial information from its tree structure and a given subset of its polynomials (see [8] for more details).

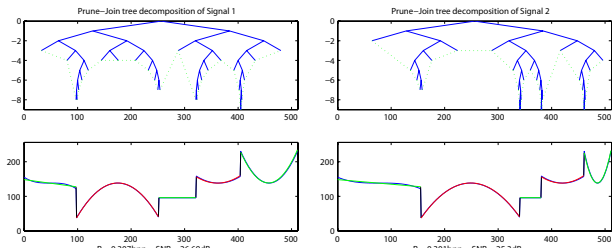


Fig. 2. Joint-Prune binary tree decomposition of two piecewise polynomial signals with shifted discontinuities.

## 3. DISTRIBUTED COMPRESSION OF MULTI-VIEW IMAGES USING 2D QUADTREE DECOMPOSITION

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

The prune-join binary tree decomposition algorithm that we used in the 1D case has an intuitive extension in 2D. We give here a sketch of our implementation (Algorithm 1) of the quadtree compression approach proposed in [9]. Figure 3(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 3(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>).

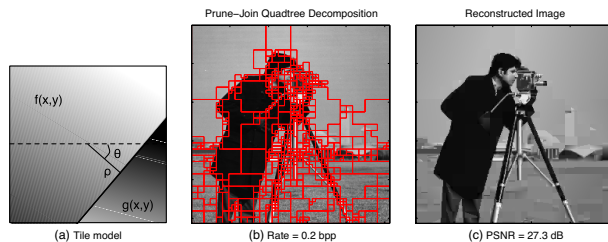


Fig. 3. (a) Our geometrical model consists of two 2D linear regions separated by a 1D linear boundary. This boundary is represented with two coefficients  $(\theta, \rho)$  and each 2D piece is represented with three coefficients  $(c_1, c_2, c_3)$  such that  $f(x, y) = \sum_{i=1}^3 c_i \mathcal{L}_i(x, y)$ , 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 1** 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 3(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:  
$$\sum_{i=1}^4 (DC_i + \lambda RC_i) \geq (D_p + \lambda R_p).$$
  - 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).
- 

### 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 can be 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 polynomials<sup>1</sup> 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 Section 3.1, 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 leaf. 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.

---

<sup>1</sup>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.

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.

### 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 [8]. 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 views satisfy exactly our piecewise polynomial model 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 an error by matching this discontinuity to another one, which would lead to a bad reconstruction.

In order to fix this problem, we can transmit some extra (redundant) information to help the decoder perform a correct matching. In the next section, we present some simulation results obtained on real multi-view images where the encoders transmit their full quadtree structure along with some information about the constant part of their polynomials to help the decoder perform a correct matching.

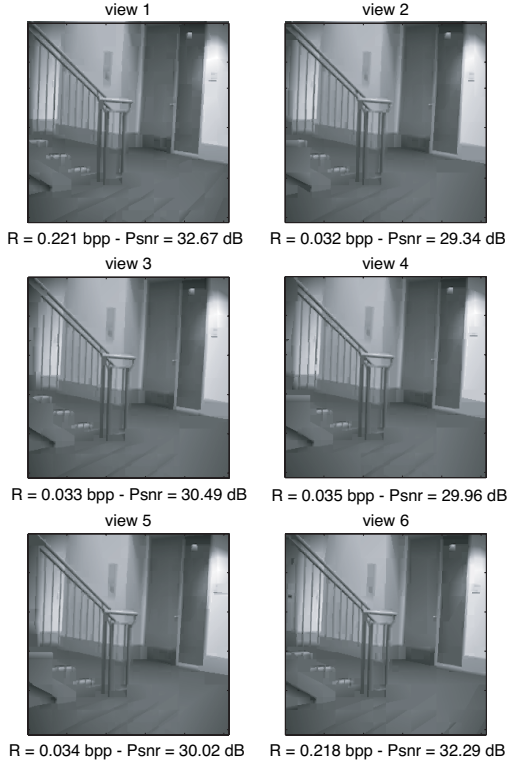
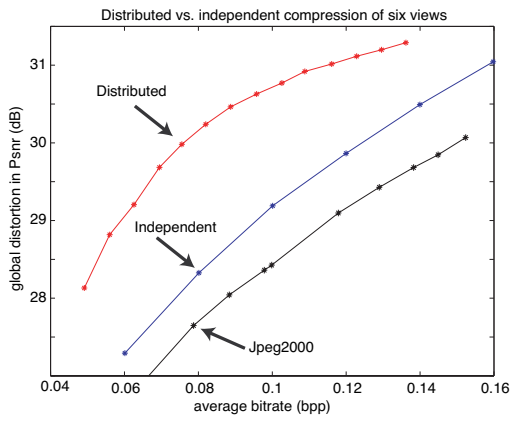
## 4. SIMULATION RESULTS

We present some simulation results obtained on the *lobby* multi-view sequence from Shum et al. [1]. Figure 4 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 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.

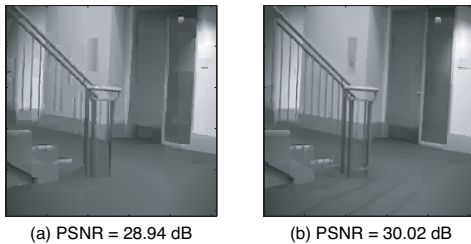
Figures 5 and 6 show results obtained on a sequence of six different views, where the first and the sixth views are fully transmitted and only the quadtree structure is transmitted for the other four views. The graph at the top of Figure 5 shows that our approach outperforms an independent encoding of the six views for all the range of considered bit-rates.



**Fig. 4.** 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 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.



**Fig. 5.** Distributed vs. independent encoding of six views. The plot at the top also shows results obtained using a Jpeg2000 encoder. The six images shown are the result of a distributed encoding with an average bit-rate of 0.08 bpp.



**Fig. 6.** Reconstruction of view 5. (a) Independent encoding at 0.08 bpp. (b) Distributed encoding with an average of 0.08 bpp.

## 5. CONCLUSIONS

We have proposed a distributed compression approach for multi-view images based on a geometrical quadtree decomposition scheme. Our approach allows for a flexible allocation of the bit-rates amongst the encoders and have been shown to outperform independent encoding even when the correlation model is not fully respected. Future work will focus on the extension to more complex multi-camera set-ups and correlation models.

## 6. REFERENCES

- [1] H.-Y. Shum, S.B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, November 2003.
- [2] M. Levoy and P. Hanrahan, "Light field rendering," in *SIGGRAPH, Computer Graphics Proceedings*, May 1996.
- [3] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Info. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.
- [4] A.D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Info. Theory*, vol. 22, no. 1, pp. 1–10, January 1976.
- [5] 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.
- [6] 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.
- [7] N. Gehrig and P.L. Dragotti, "Distributed compression of the plenoptic function," in *IEEE ICIP '04*, October 2004.
- [8] N. Gehrig and P.L. Dragotti, "DIFFERENT - Distributed and Fully Flexible image EncodeRs for camEra sensor NeTworks," in *IEEE ICIP '05*, September 2005.
- [9] R. Shukla, P.L. Dragotti, M.N. Do, and M. Vetterli, "Rate-distortion optimized tree structured compression algorithms for piecewise polynomial images," *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 343–359, March 2005.
- [10] X. Zhu, A. Aaron, and B. Girod, "Distributed compression for large camera arrays," in *IEEE SSP '03*, September 2003.
- [11] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compression of light-field rendered images using coset codes," in *37th Asilomar Conference on Signals, Systems, and Computers: Special Session on Distributed Coding*, November 2003.
- [12] 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 *EU-SIPCO '05*, September 2005.
- [13] B. Song, O. Bursalioglu, A.K. Roy-Chowdhury, and E. Tuncel, "Towards a multi-terminal video compression algorithm using epipolar geometry," in *IEEE ICASSP '06*, May 2006.
- [14] C. Yeo and K. Ramchandran, "Robust distributed multi-view video compression for wireless camera networks," in *SPIE Visual Communications and Image Processing (VCIP'07)*, January 2007.