Balanced Distributed Coding of Omnidirectional Images

Vijayaraghavan Thirumalai, Ivana Tosic and Pascal Frossard *

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Institute, Lausanne, 1015 - Switzerland.

ABSTRACT

This paper presents a distributed coding scheme for the representation of 3D scenes captured by stereo omnidirectional cameras. We consider a scenario where images captured from two different viewpoints are encoded independently, with a balanced rate distribution among the different cameras. The distributed coding is built on multiresolution representation and partitioning of the visual information in each camera. The encoder transmits one partition after entropy coding, as well as the syndrome bits resulting from the channel encoding of the other partition. The decoder exploits the intra-view correlation and attempts to reconstruct the source image by combination of the entropy-coded partition and the syndrome information. At the same time, it exploits the inter-view correlation using motion estimation between images from different cameras. Experiments demonstrate that the distributed coding solution performs better than a scheme where images are handled independently, and that the coding rate stays balanced between encoders.

Keywords: Distributed source coding, Omnidirectional images, LDPC codes

1. INTRODUCTION

Vision sensor networks find widespread usage in several applications like 3DTV, surveillance, object tracking etc. to collect and process data from the monitoring environment. These sensor networks normally consist of large number of camera sensors distributed over an area, to capture the same scene in different view points. There are several interesting problems in this scenario e.g., compression, reconstruction of the 3D structure from multiple views, object recognition, etc. This paper paper focuses on compression of the multi-view images, and in particular considers the case of two cameras (e.g., stereo vision). The correlation between the multiview images could be easily exploited by joint encoding using motion estimation or disparity compensation. The joint encoding approach is for example employed in the Multiview Video Coding (MVC) architecture for the 3DTV concept,[1] by performing motion estimation both in the temporal and inter-view domain. This obviously achieves better performance over the independent coding solution. However, distributed compression is generally preferred in vision sensor networks, because it avoids inter-communication between the sensors.

The first information-theoretical results on distributed source coding appeared in the late seventies for the noiseless[2] and noisy case.[3] In particular, it has been shown[2] that independent coding of correlated sources can achieve the same rate-distortion bound as joint encoding if a joint decoder can efficiently exploit the correlation between the sources. However, most results in distributed coding have remained non-constructive for about three decades. Practical DSC schemes have been designed only recently, by establishing a relation between the Slepian-Wolf theorem and channel coding.[4] Subsequently, several practical DSC systems have been presented using different channel codes. e.g., Turbo code,[5,6] LDPC code,[7] Arithmetic code[8] etc. Later this concept has been used to construct video coders,[9,10] where the frames are encoded independently and decoded jointly. This brings down the complexity of the encoder, by shifting the motion estimation (ME) to the joint decoder. Few studies have been reported in the literature on application of distributed coding of the camera networks. In these scenarios, researchers have reached improved coding performance by exploiting the correlation between the images by (i) disparity estimation,[11,12,16] (ii) motion estimation[13,14] and (iii) coset encoding with optimal design of quantizers.[15] The problem of compressing the multi-view video using DSC principles are presented in the references.[28,29] DSC technique has been also applied to compress the hyperspectral images.[17] Most of these works target a particular problem in distributed coding, that is coding with side information. In this case, one

* This work has been partly supported by the Swiss National Science Foundation, under grant 20001-107970/1.

Further author information:{vijayaraghavan.thirumalai, ivana.tosic, pascal.frossard}@epfl.ch

image is used as side information for other images that have been encoded along distributed coding principles. In this case, a hierarchical relation is introduced among frames, and the coding rates are usually unbalanced.

In practice however, it is often interesting to avoid hierarchical relations between sensors and to distribute the coding and transmission cost equally among the sensors. Only few works have considered the problem of similar rate allocation among the sensors. Early work on symmetric coding focusses on time sharing.[21] However, the inter-source synchronization to perform time sharing is hard to implement, especially between the non-communicating encoders. The first practical scheme for symmetric coding for was proposed in[22] based on channel code partitioning. Subsequently Stankovic *et al.*[23] extended the scheme to multiple sources using systematic channel codes. Sartipi *et al.*[24] considered the compression of two sources at the symmetric rate by information partitioning where half of the source bits are transmitted directly while the corresponding syndrome bits are generated on the other half (complementary part) of the source bits. In the context of distributed compression in camera networks, a balanced DSC scheme was proposed in[25] , based also on the information partitioning approach. However,[25] differs from the work presented in this paper in two aspects. First,[25] focuses on a Slepian-Wolf code design and does not offer a practical design for balanced multi-view image coding. Second, it requires the knowledge of camera parameters and camera calibration, which is not necessary for the approach presented in this work.

We propose a rate-balanced distributed coding scheme for spherical stereo,[26] where the omnidirectional images are captured by the catadioptric sensors. The catadioptric camera is realized by a convex, reflective parabolic mirror placed above and parallel to the camera approximating an orthographically projecting lens. In such a case, the ray of light incident with the focus of the parabola is reflected to a ray of light parallel to the parabola's axis. This construction is equivalent to a purely rotating perspective camera. The omnidirectional images are mapped on the sphere which captures the plenoptic function in its radial form without discrepancies that usually appear due to Euclidian assumptions for planar images. Compared to perspective stereo, spherical stereo certainly provides great advantage and flexibility in representing the 3D scene. We design a transform-based distributed coding algorithm, where omnidirectional images initially undergo a multi-resolution decomposition. The visual information is then split into two correlated partitions. The first partition is entropy coded and sent to the decoder. Channel coding is applied to the second partition, and only the syndrome bits are sent to the decoder. The joint decoder uses motion estimation to take benefit of the correlation between views, and uses the syndrome bits to decode the missing information. Therefore, the proposed scheme represents an interesting combination of the intra and inter Wyner-Ziv image coding that permits to balance the rate between cameras. Such a strategy proves to be beneficial with respect to independent processing of the omnidirectional images, and shows only a small performance loss compared to joint encoding of the different views. Finally, exploiting the inter-view correlation by motion estimation is highly advantageous in camera network setups where no knowledge of the camera parameters is given or when camera network calibration is not achievable in practice.

The rest of the paper is organized as follows. Section 2 proposes an overview of the distributed coding scheme. Section 3 presents the encoder in details, and Section 4 describes the joint decoding scheme. Section 5 describes the implementation details and presents and discusses the simulation results. Section 6 concludes this paper.

2. SYMMETRIC DISTRIBUTED CODING SCHEME

This work focuses on the distributed compression for stereo cameras (Fig. 1), where the images X and Y are captured by the catadioptric sensors C_x and C_y respectively. Under this circumstances, the spherical images X and Y are likely to be correlated due to overlapping views. Each camera output is compressed independently and is processed at the joint decoder to improve the performance by exploiting the correlation between the views.

In this setting, we are interested in balancing the transmission rate between the encoders. The progressive encoder (Fig.1) provides multi-resolution representation of the spherical image (e.g., wavelet transform), followed by the bit plane or zero-tree encoding (e.g., SPIHT coding). We propose to exploit the correlation between neighboring coefficients by splitting the information into two correlated partitions. Hence, each encoder transmits one half of the bits from the bitplanes along with the corresponding syndrome bits generated using channel coding on the rest of the source bits.

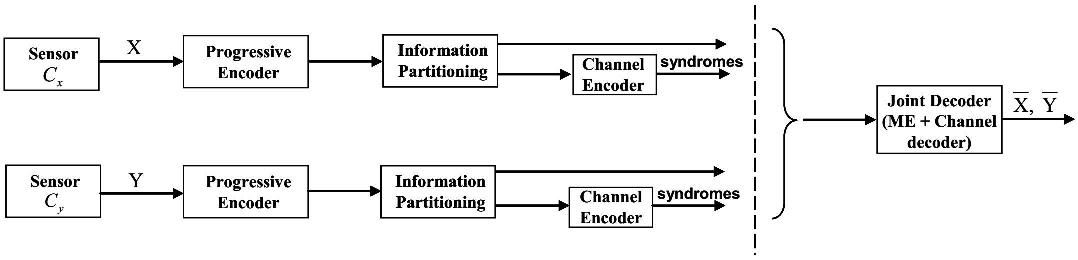
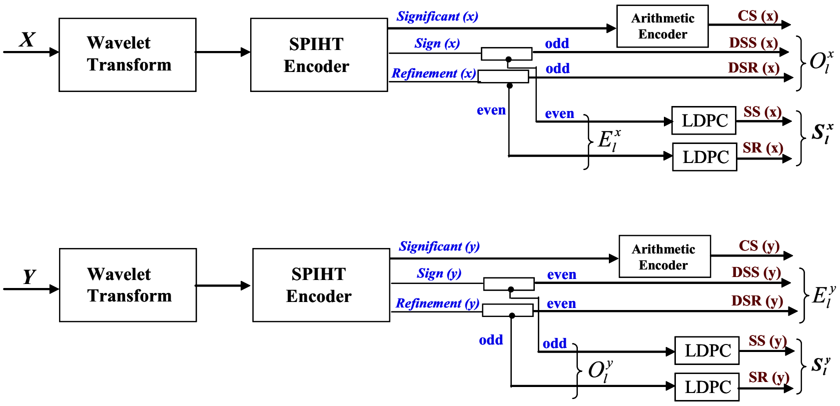


Figure 1. System for compressing the correlated spherical images using symmetric DSC

The joint decoder progressively decodes the bit plane starting from the MSB to the LSB. For each bit plane, as one of the correlated partitions is intra coded, we propose to predict the visual information in other partition, by exploiting the correlation between the partitions. Under the assumption that the images X and Y are correlated, the predicted results are further refined using the motion estimation (ME) performed on the sphere,[20] in order to compensate the displacement of the objects captured from different view points. Thus, our approach generates the side information by efficiently exploiting both intraview and interview correlation using bit prediction and the ME steps respectively. Subsequently, the bit plane is recovered after correcting the virtual channel noise in the side information using the syndromes. Thus, the joint decoder recovers the bit planes successively and the images are progressively reconstructed.

3. ENCODER



CS: Compressed significant bits SS: Sign syndrome bits RS: Refinement syndrome bits
DSS: Directly sent sign bits DSR: Directly sent refinement bits

Figure 2. Block schemes of the independent encoders.

The encoding strategy is now discussed in more details. As shown in Fig. 2 the unwrapped images X and Y are decorrelated using the Wavelet transform and then encoded using SPIHT.[18] The benefit of SPIHT coding is that it brings good compression performance with little processing overhead. Here we briefly discuss how SPIHT works and for further details we refer the reader to the reference.[18] SPIHT uses a significancy test to identify the significant coefficients and progressively encodes them starting from MSB to LSB bit planes. A coefficient c is said to significant in the bitplane l if $2^l \leq \ |c| \ < 2^{l+1}$. The algorithm is initialized by setting the list of significant pixels (LSP) to an empty list, tree roots to the list of insignificant pixels (LIP) and the descendants to the list of insignificant sets (LIS). The progressive bit plane encoding is achieved by means of two coding passes.

- Sorting pass : Significancy test is applied to the coefficients in the LIS and LIP, to determine whether they are significant or not in the bit plane l. Those coefficients which are found to be significant are moved to the LSP. Furthermore, the corresponding sign bits of the significant coefficients are outputted.

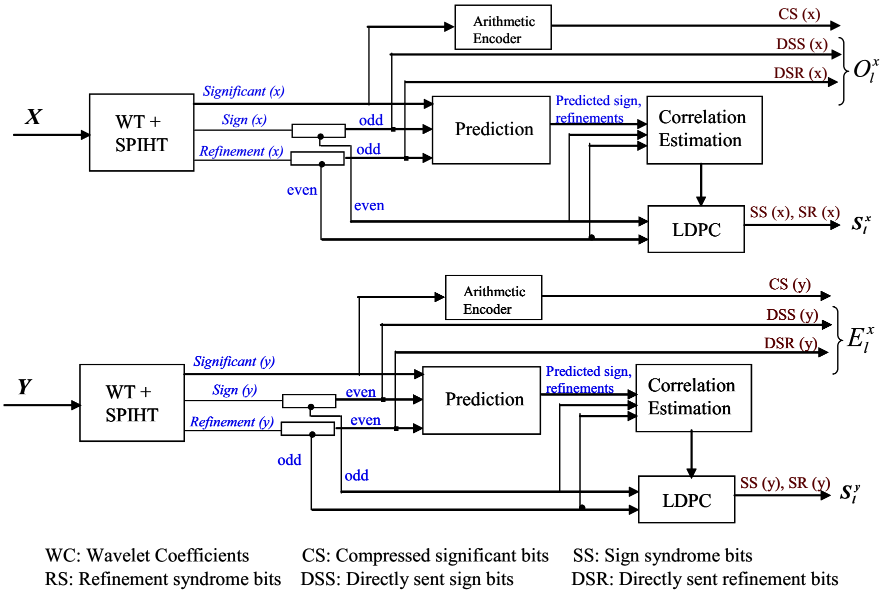


Figure 3. The modified encoding scheme with correlation estimation

- Refinement pass: All the coefficients in the LSP, except for those coefficients added in the last sorting pass (i.e., with same l) are visited again to output the l^{th} bit of the coefficient.

We denote the bit plane l generated from the image X by $B_l^x = [b_l^x(1), b_l^x(2), ...b_l^x(n)]$, where $b_l^x(i)$ represents the i^{th} bit in the bit plane l and n represents the length of the bit plane. The even and odd indexed bits in the bit plane l are represented by $E_l^x = [b_l^x(2), b_l^x(4), ...]$, $O_l^x = [b_l^x(1), b_l^x(3), ...]$ respectively. In a similar way, we extend our notations to the image Y. The generated significant bits are entropy coded and are eventually used by the decoder to identify the significant coefficients, while the sign and the refinement bits are encoded by Slepian-Wolf coding. We then subsample each bitplane l [†], such that the source bits at odd positions O_l^x in image X, and bits at even positions E_l^y in image Y are directly transmitted. Instead of transmitting the rest of the source bits E_l^x and O_l^y, we generate and transmit only the syndrome bits S_l^x and S_l^y after channel coding.

Channel coding is implemented with irregular Gallager LDPC codes and we follow the procedure suggested by Neal[19] to construct the LDPC parity check matrix. We further model the virtual channel as a Binary Symetric Channel (BSC) with capacity C. The crossover probability p is estimated by comparing the original bit plane and the corresponding side information bit plane. In other words, the rate of the LDPC code is computed by taking into account the correlation between neighboring bits and the efficiency of the motion estimation that can be computed offline on test images. We denote as E the error in the predicted bitplane and E' the error in the side information bitplane, calculated with respect to the original bitplane. The bit prediction step is carried out at the encoder itself to obtain E and then we estimate the portion of error that can be corrected by the ME, evaluated by $\delta E = \frac{E-E'}{E}$. The parameter δE quantifies the benefit of ME and is determined heuristically using several image sets. By knowing δE and the size of the bitplane l, the crossover probability is calculated using the following expression:

$$p = \frac{E - \lfloor \delta E \times E \rfloor}{l/2}. \tag{1}$$

In the evaluation of p, $l/2$ bits are considered because half of the bit plane is encoded using LDPC codes. Once p is estimated, the rate R of the LDPC code is selected such that $R < C$, where $C = 1 - H(p)$. The modified encoding scheme with correlation estimation is shown in the Fig.3.

[†] Here the bitplane l refers to the corresponding bitplane l of both the sign and the refinement bits.

4. JOINT DECODER

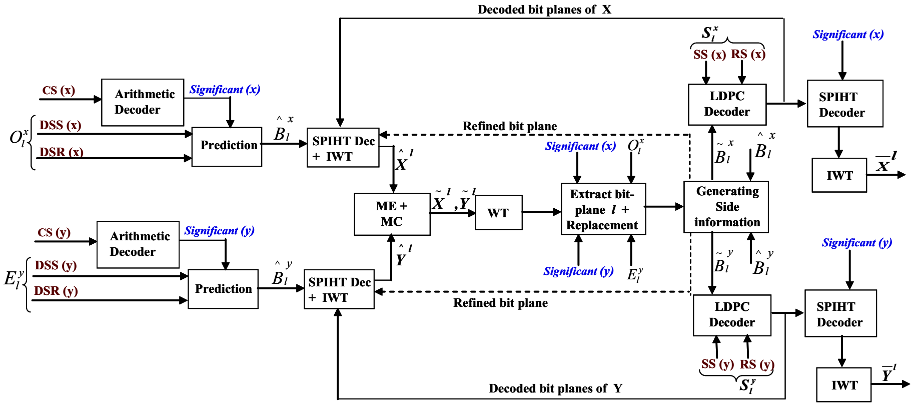


Figure 4. Block scheme of the joint decoder.

The proposed joint decoding scheme is illustrated in the Fig. 4. The joint decoder reconstructs the bit planes progressively starting from MSB to LSB. Also, the bit planes of the images X and Y that are successfully decoded are used to refine the side information for decoding the following bit planes of X and Y. Assume that the first $l-1$ bit planes of X (i.e., B_k^x for $1 \leqslant k \leqslant l-1$) are successfully decoded and the image reconstructed using $l-1$ bit planes is represented by $\bar{X}^{l-1}$. Similarly $\bar{Y}^{l-1}$ is reconstructed using the $l-1$ decoded bit planes. Now, the joint decoder reconstructs the bit planes B_l^x and B_l^y by decoding the corresponding unknown bits E_l^x and O_l^y in the following steps:

- Prediction: Unknown bits E_l^x and O_l^y are predicted from their corresponding deterministic bits O_l^x and E_l^y respectively.
- Motion Estimation: Side information is generated by refining the predicted bits.
- LDPC decoder: Virtual channel noise is corrected in the side information using syndrome bits.

4.1 Prediction

The objective of the prediction is to predict the unknown bits E_l^x in the image X and O_l^y in the image Y from their corresponding deterministic bits O_l^x and E_l^y respectively, which are transmitted directly to the joint decoder. Fig.5 illustrates the scenario for the image X, where the predictor takes the deterministic bits O_l^x as the input (marked in black colour). The blank spaces (E_l^x) between the deterministic bits are the unknown bits to be predicted. We denote the predicted bit plane l by $\hat{B}_l^x$ and the predicted bits are marked in gray colour. Similarly for the image Y, the unknown bits O_l^y are predicted from the corresponding deterministic bits E_l^y and the predicted bit plane l is represented by $\hat{B}_l^y$.

Let us discuss now how the unknown bits are predicted from the corresponding deterministic bits. Initially, the coordinates of the deterministic bits and the unknown bits in the subband decomposition are decoded using the corresponding significant tree information. Once the coordinates of the unknown bits are known, we predict them using the nearest significant neighbors. For an unknown bit at the coordinate (ϕ, θ) there are eight nearest neighbors that form the set NN, among which a subset of significant neighbors $SN \subseteq NN$ are selected as the candidates for the prediction. Significant neighbors SN are selected from the set NN by looking for nearest neighbors containing a deterministic bit. Once the nearest significant neighbors are known, the unknown bit is predicted by performing majority vote among these candidates. For special cases where the number of ones equals the number of zeros within SN, or no significant neighbors ($SN = \{\}$) are found, the value of the bit is determined by the most probable bit in the bit plane, calculated from the distribution of the corresponding deterministic bits.

In more details, let i be the bit to be predicted, present in the LL band (Fig.6). The nearest neighbors for the bit i are $NN = \{a, b, c, d, e, f, g, h\}$. Among these neighbors, we select the significant neighbors by looking

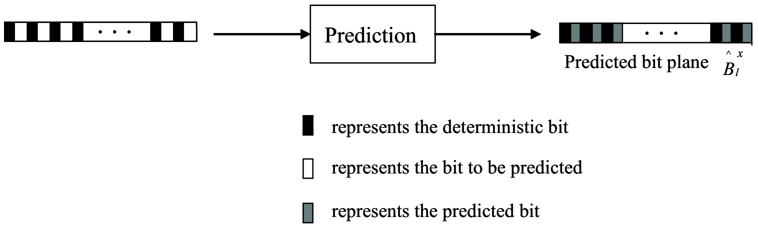

Prediction
Predicted bit plane $\hat{B}_l^x$
represents the deterministic bit
represents the bit to be predicted
represents the predicted bit

Figure 5. Predicting the unknown bits E_l^x in the bit plane l from the corresponding deterministic bits O_l^x. The predicted bit plane is represented by $\hat{B}_l^x$

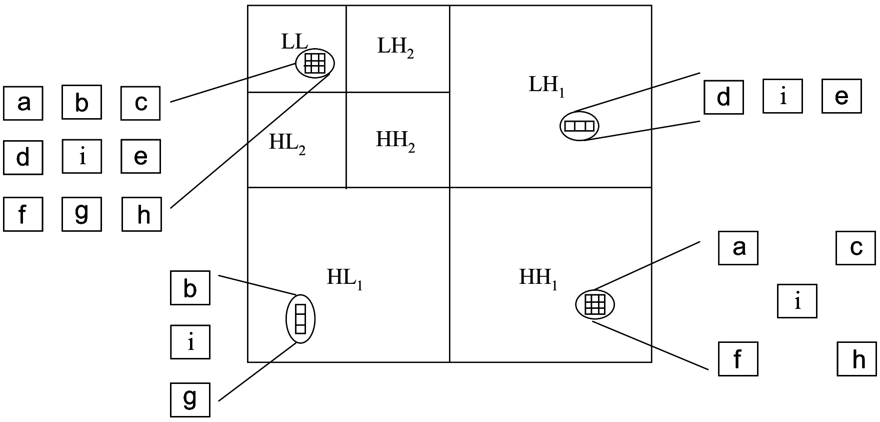


Figure 6. Prediction using the nearest significant neighbors. The bit i has to be predicted.

for the nearest neighbor containing a deterministic bit. As an example, consider $SN = \{a, c, d, f, h\}$ (i.e., the deterministic bits are found at the positions a, c, d, f and h) and the value at these positions are $\{1, 0, 1, 1, 0\}$ respectively. Now the bit i is predicted as one, because it wins the majority vote among the candidates. The motivation behind the prediction is to take benefit from the intra-correlation as there exists correlation among the neighboring coefficients, particularly in the LL band. Obviously generating bit planes from the correlated coefficients (e.g. SPIHT), also outputs correlated bits in each bit plane. In particular, the correlation among the bits in the MSB bit plane is higher and the correlation reduces from the MSB bit plane to LSB bit plane. As a result, the prediction error is smaller towards the MSB bit plane and the error increases from the MSB to the LSB bit plane.

The orthogonal property of the Wavelet filter is taken into account to predict the bits in the detail subbands. In the multi-resolution representation the edges in the LH,HL and HH subbands are oriented as horizontal, vertical and diagonal respectively. For the detail subbands, instead of selecting significant neighbors from the set NN, we select a subset of nearest neighbors $NN' \subset NN$ and then the significant neighbors from NN'. The nearest neighbors NN' are carefully selected from the set NN based on the orientation of the edge in the respective detail subband. For example, for the unknown bit i whose coordinate lies in the LH band, the SN are selected from $NN' = \{d, e\}$, by assuming d, i and e form a horizontal edge. Similarly, we predict the unknown bit i in the HL and HH sub bands using the significant neighbors selected from the nearest neighbors $NN' = \{b, g\}$ and $NN' = \{a, h, f, c\}$ respectively.

4.2 Motion Estimation: Generating side information

This section discusses the generation of the side information from the predicted bit plane, by taking advantage from the intra view correlation. As we discussed earlier, for the bit plane l half of the sign and the refinement bits are transmitted directly from the corresponding encoders and the unknown bits are predicted from the corresponding deterministic bits. The decoded $l-1$ bit planes of the image X (B_k^x for $1 \leqslant k \leqslant l-1$) and the predicted bit plane $\hat{b}_l^x$ are combined at the SPIHT decoder. Furthermore, Inverse Wavelet transform (IWT) is applied on the result of SPIHT decoding to generate the image $\hat{X}^l$. In a similar way, the image $\hat{Y}^l$ is reconstructed from the corresponding decoded $l-1$ bit planes and the predicted bit plane $\hat{B}_l^y$. The predicted bit planes are refined by exploiting the correlation between views using ME on the sphere[20] and it is performed between the approximated images $\hat{X}^l$ and $\hat{Y}^l$.

The motion vectors are generated using multi-resolution ME, in order to minimize the computational cost. We explain here how the ME on the sphere works, while more details can be found in the reference.[20] Initially multi-resolution representations of the images $\hat{X}^l$ and $\hat{Y}^l$ are computed and we denote $\hat{X}_L^l$ and $\hat{Y}_L^l$ be the low resolution spherical images respectively. The low resolution spherical image $\hat{X}_L^l$ is divided into blocks of uniform solid angles. For each block in $\hat{X}_L^l$, a best matching block in the mean square sense is selected in $\hat{Y}_L^l$. The displacement between the blocks represents the motion vector. The generated motion vectors from the lower resolution serve as the initial estimate for the higher resolution. The initial estimate is further refined using the spherical images at the higher resolution. This process is iterated up to the finest resolution and eventually outputs the motion vectors. The generated motion vectors are then used to compensate the motion in the image $\hat{Y}^l$ and the motion compensated (MC) image is represented by $\tilde{X}^l$.

The motion compensated images $\tilde{X}^l$ and $\tilde{Y}^l$ further undergo wavelet transform (same transformation used on X and Y) and the corresponding bit plane l is extracted using the significance bits of X and Y respectively. As half of the bits in the bit plane l are transmitted directly, we directly replace the bit value at the corresponding bit index with the deterministic bits. The ME captures the correlation efficiently in the LL band and not in the detail subbands, and thus bits that were predicted in the LL band are only refined with ME. So due to the high pass characteristics of the ME algorithm, we generate the side information after combining the ME and the bit prediction results. More precisely, the side information is generated by combining the bits in the LL band that are refined by the ME and bits in the detail bands that are predicted using the bit prediction step.

It is clear that, at this stage the error (w.r.t corresponding original bit plane) in the side information bit plane is certainly smaller than the prediction error E, as the inter-correlation is exploited in the former case and not in the later case. As the bit planes are refined now, obviously we have better knowledge about the correlation between the views and we found experimentally that it is beneficial to repeat the ME step once again. Hence, we perform the ME step once again between the approximated images $\hat{X}^l$ and $\hat{Y}^l$, where the images $\hat{X}^l$ and $\hat{Y}^l$ are generated by combining the corresponding $l-1$ decoded bit plane and the refined bit plane l (marked in dotted line in the Fig. 4). After the ME step the error would be reduced further, due to the better knowledge of interview correlation and the side information ($\tilde{B}_l^x$ and $\tilde{B}_l^y$) is generated by combining the result of ME and the prediction, as explained above.

It can be noted that the decoding complexity can be reduced by skipping the ME for the most significant bitplanes. The bit prediction step is quite efficient for these bitplanes, such that the first n bitplanes can be decoded without activating the ME step. However, for the least significant bitplanes, the error due to prediction increases and refinement by ME is able to take advantage of the correlation between the images.

4.3 LDPC decoding

LDPC decoder uses the generated side information bit planes $\tilde{B}_l^x$ and $\tilde{B}_l^y$ and the syndrome bits S_l^x and S_l^y to decode the bit planes B_l^x and B_l^y respectively. LDPC decoder uses Belief propagation algorithm where the confidence level of each variable node v_i is initialized using the log likelihood ratio (LLR) as given by the following equations:

$$L(v_i) = log\left(\frac{P(b_l^x(2i)=0|\tilde{b}_l^x(2i)}{P(b_l^x(2i)=1|\tilde{b}_l^x(2i)}\right) = (1-2\tilde{b}_l^x(2i))log(\frac{1-p}{p}), \tag{2}$$

$$L(v_i) = log\left(\frac{P(b_l^y(2i-1)=0|\tilde{b}_l^y(2i-1)}{P(b_l^y(2i-1)=1|\tilde{b}_l^y(2i-1)}\right) = (1-2\tilde{b}_l^y(2i-1))log(\frac{1-p}{p}), \tag{3}$$

where p denotes the cross over probability of BSC.

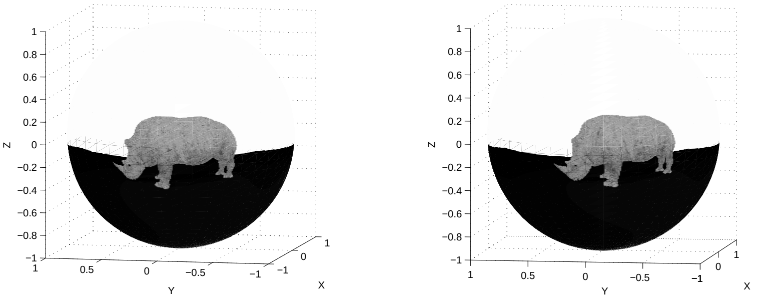


Image X **Image Y**

Figure 7. Original Rhino images

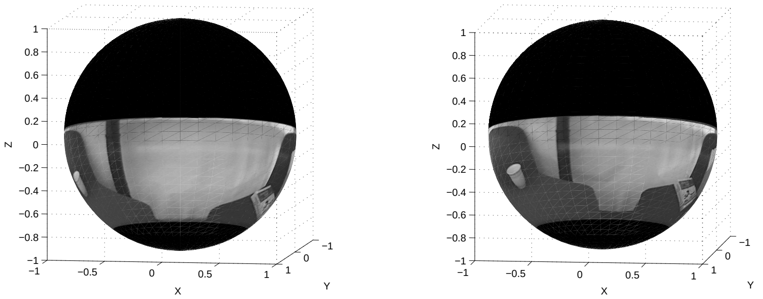


Image X **Image Y**

Figure 8. Original Lab images

It should be noted that the LDPC decoding is carried out on the odd samples for source X and on even samples for source Y, as their complementary samples are intra coded. The bits $\tilde{E}_l^x$ ($\tilde{O}_l^y$) in the side information bit plane $\tilde{B}_l^x$ ($\tilde{B}_l^y$) are visualized as the noisy version of the corresponding original bits E_l^x (O_l^y). Mathematically, this is represented as follows :

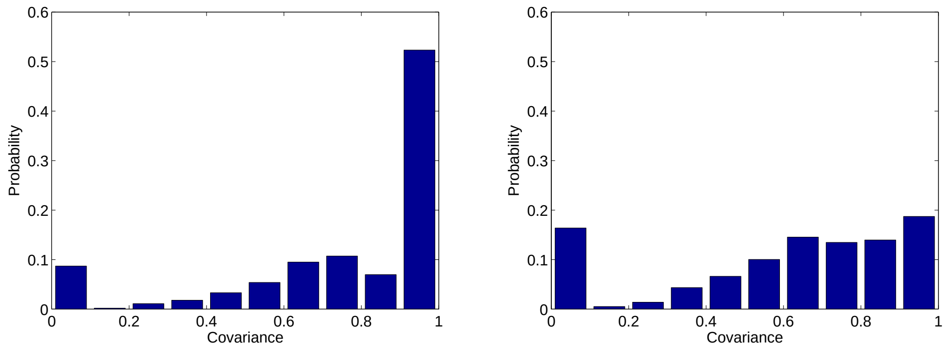
$$\tilde{E}_l^x = E_l^x \oplus Z^x, \tag{4}$$

$$\tilde{O}_l^y = O_l^y \oplus Z^y, \tag{5}$$

where Z^x (Z^y) is the noise that consists of zeros, except for some ones that mark the position where $\tilde{E}_l^x$ and E_l^x ($\tilde{O}_l^y$ and O_l^y) differ. The virtual channel noise is corrected by using the syndrome bits S_l^x (S_l^y) at the LDPC decoder. When the refinement bits are decoded, they are added to the corresponding coefficients using the significant tree of X and Y and the decoder applies inverse wavelet transform to reconstruct the images $\bar{X}^l$ and $\bar{Y}^l$. The successfully decoded l^{th} bit plane is used to decode the further bit planes as explained previously.

5. EXPERIMENTAL RESULTS

The proposed distributed coding system has been implemented with a multiresolution decomposition of the unwrapped spherical images, based on Daub 9/7 wavelet filters with three levels of decomposition. Multiresolution motion estimation is implemented with blocks of size 4x4. In the joint decoder, the motion estimation is only activated after the decoding of the first two bitplanes. The performance of the system has been evaluated on synthetic and natural image sets respectively the Rhino (Fig. 7) and Lab images (Fig. 8).



(a) **(b)**

Figure 9. Distribution of the covariance values calculated for the wavelet coefficients in the LL band (a) Rhino Image X (b) Lab Image X (evaluated without the dark portion of the image)

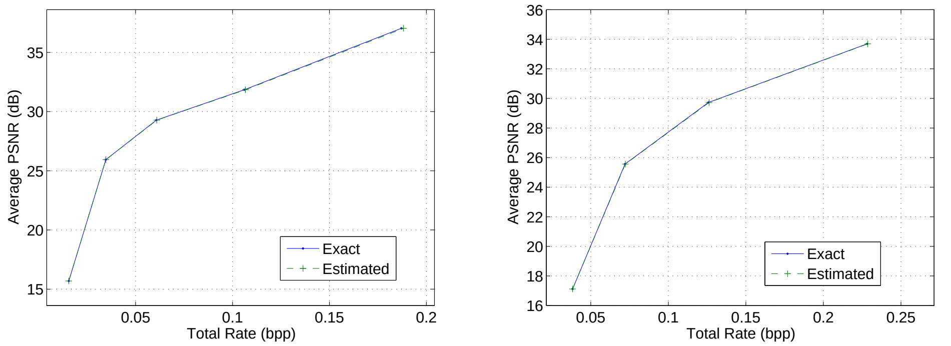
We first calculate the covariance value among neighboring coefficients since it motivates the design of the coefficient prediction in the distributed coding algorithm. Fig. 9 shows the histogram of the covariance values of the coefficients in the LL band, for the Rhino and Lab images. A significant percentage of the coefficients has high covariance values. Additionally, one can show that correlation mostly lies along the horizontal, vertical and diagonal directions in the LH, HL and HH subbands, respectively. We then evaluate the benefit of using ME on the joint decoder and we observe that the ME further reduces the bit error by 8-18 % (w.r.t the predicted bit plane) on the test images. The crossover probability of the channel is calculated using the equation 1, by fixing $\delta E = 0.08$. However it should be noted that the p for the first two bit planes are calculated using $p = \frac{E}{l/2}$, as these bit planes are decoded with out activating ME. In order to analyze the loss in decoding performance due to the imprecise channel rate estimation, we have compared rate-distortion (RD) performance between our estimated model and exact model in which the exact value of the p is known at the encoder. The comparison is available in the Fig. 10 and we observe that the degradation in RD performance is negligible, up to 0.08 db loss on tested images.

The RD performance of our scheme has been compared to a SPIHT-based coding scheme that handles images independently, both at the encoder and at the decoder. In the independent coding scheme the significant bits are entropy coded, in order to highlight the gain due to DSC. Fig.11(a) and (b) show the RD curves for the Rhino and Lab image sets, respectively. The DSC scheme outperforms independent coding scheme, where the coding gain can reach 4.5 dB (resp. 2.7 dB) at low rate and 2.5 dB (resp. 1 dB) at high rate. The decreasing gap at high rate is mostly due to the fact that the bitplanes towards the LSB position are less correlated. The gain is higher for Rhino images as they are more correlated than the Lab images. Interestingly, we see also that the encoding is quite balanced, since the encoding rates for images X and Y are quite similar for a given reconstruction quality.

We further evaluate the performance of the distributed coding scheme compared to a joint encoding strategy, as shown in Fig. 12. The joint encoding scheme codes one image as a reference frame with a SPIHT-based strategy. Multiresolution ME is used to predict the second image, and the residual error after motion estimation is also encoded using SPIHT. The motion vectors at the lowest resolution are transmitted first and the refinements between the successive resolution are then transmitted. Thus, the motion vectors are transmitted progressively, and the rate allocation at the joint encoder is chosen such that the rate-distortion performance is maximized. We observe that the distributed coding scheme performs similarly to the joint encoding algorithm that is based on the same representation and progressive coding functions.

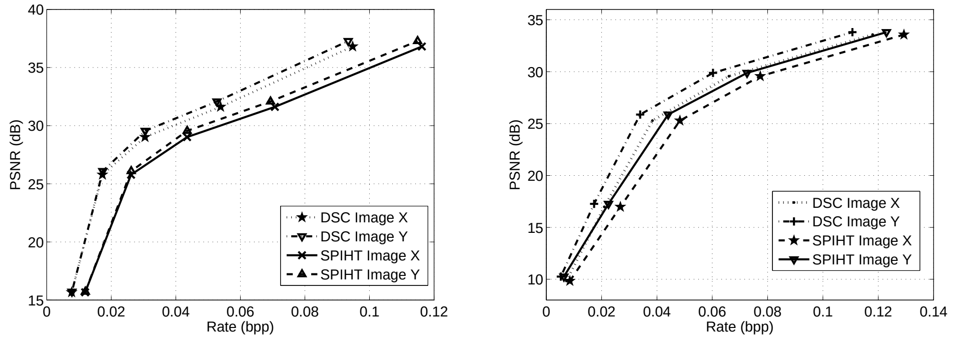
6. CONCLUSIONS

In this paper we have presented a balanced transform-based distributed source coding scheme for the representation of scenes with camera networks. It efficiently exploits both the intra- and interview correlation by coefficient



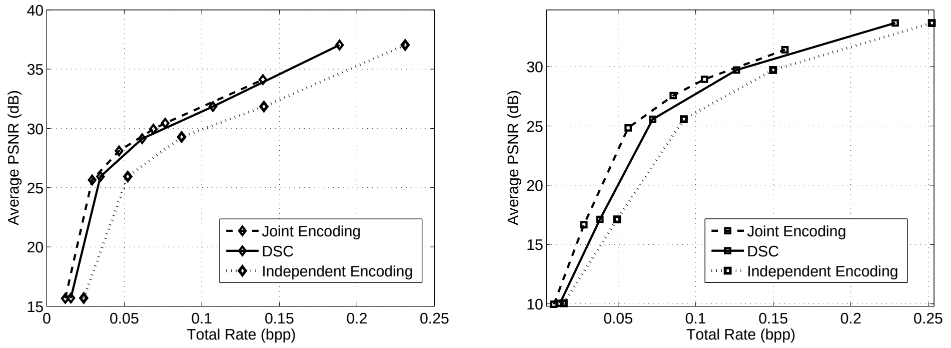
(a) **(b)**

Figure 10. RD comparison between the estimated and the exact correlation models (a) Rhino Images (b) Lab Images



(a) **(b)**

Figure 11. RD comparison between the proposed DSC and SPIHT coding schemes (a) Rhino Images (b) Lab Images



(a) **(b)**

Figure 12. RD comparison between the proposed DSC, SPIHT and Independent coding schemes (a) Rhino Images (b) Lab Images

prediction and motion estimation respectively at the decoder. Our scheme outperforms the independent coding and performs close to a joint encoding solution based on similar coding principles. Finally, it should be noted that the rate of the LDPC coder has been chosen based on a conservative estimation on sets of images with limited correlation. However, we have observed only a negligible degradation in rate-distortion performance with respect to the oracle based scheme. Our proposed scheme is limited to two cameras, and our future work focuses on developing a symmetric DSC scheme for multiple cameras.

REFERENCES

1. P. Merkle, A. Smolic, K. Muller and T. Wiegand, "Efficient Prediction Structures for Multiview Video Coding," IEEE Trans. on Cir. and Sys. for Video Tech., vol.17(11), pp. 1461-1473, Nov. 2007.
2. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources", IEEE Trans. on Inform. Theory, vol. 19(4), pp. 471-480, July 1973.
3. A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side-information at the decoder", IEEE Trans. on Inform. Theory, vol. 22(1), pp. 1-10, Jan. 1976.
4. S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS)", IEEE Trans. on Inform. Theory, vol. 49(3), pp. 626-643, Mar. 2003.
5. J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," IEEE Commun. Lett., vol. 5(10), pp. 417-419, Oct. 2001.
6. A. Aaron and B. Girod, "Compression with side information using turbo codes," Proc. of IEEE DCC, pp. 252-261, Apr. 2002.
7. A. D. Liveris, Z. Xiong and C. N. Georghiades, "Compression of binary sources with side-information at the decoder using LDPC codes", IEEE Commun. Lett., vol. 6(10), pp. 440-442, Oct. 2002.
8. M. Grangetto, E. Magli and G. Olmo, "Distributed Arithmetic Coding", IEEE Commun. Lett., vol. 11(11), pp. 883-885, Nov. 2007.
9. A. Aaron, S. Rane, E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," Proc. SPIE Visual Commun. and Image Proc., pp. 520-528, Jan. 2004.
10. R. Puri, A. Majumdar and K. Ramchandran, "PRISM: A Video Coding Paradigm With Motion Estimation at the Decoder" IEEE Trans. on Image Proc. , vol. 16(10), pp. 2436 - 2448, Oct.2007.
11. X. Zhu, A. Aaron and B. Girod, "Distributed compression for large camera arrays", Proc. IEEE SSP 2003, pp. 30-33, Sept. 2003.
12. N. Gehrig and P.L.Dragotti, "Distributed Compression of Multi-View Images using a Geometric Approach", Proc. of IEEE ICIP, Sept. 2007.
13. T. Tillo, B. Penna, P. Frossard and P. Vandergheynst, "Distributed Coding of Spherical Images with Jointly Refined Decoding," pp. 1-4, Proc. of IEEE MMSP, Nov. 2005.
14. V. Thirumalai, I. Tosic and P. Frossard, "Distributed Coding of Multiresolution Omnidirectional Images," Proc. of IEEE ICIP, pp. 345-348, Sept. 2007.
15. J. Kusuma, L. Doherty and K. Ramachandran, "Distributed Compression for Sensor networks," Proc. of IEEE ICIP, pp. 82-85, Oct. 2001.
16. D. Varodayan, A. Mavlankar, M. Flierl and B. Girod, " Distributed Grayscale Stereo Image Coding with Unsupervised Learning of Disparity, " Proc. of IEEE DCC, pp.143-152, Mar.2007.
17. N.M. Cheung, C. Tang, A. Ortega and C.S. Raghavendra, "Efficient Wavelet-based Predictive Slepian-Wolf Coding for Hyperspectral Imagery," EURASIP Journal on Signal Proc., vol.86(11), pp. 3180-3195, Nov.2005.
18. A. Said and W. A. Pearlman, "A new, fast, and efficient image codec using set partitioning in hierarchical trees", IEEE Trans. on Cir. and Sys. for Video Tech., vol. 6(3), pp. 243-250, June 1996.
19. Methods for constructing LDPC codes: Available in URL http://www.cs.utoronto.ca/pub/radford/LDPC-2001-05-04/pchk.html.
20. I. Tosic, I. Bogdanova, P. Frossard and P. Vandergheynst, "Multiresolution Motion Estimation for Omnidirectional Images", Proc. of EUSIPCO 2005, Sept. 2005.
21. F.M.J. Willems, "Totally asynchronous Slepian-Wolf data compression," IEEE Trans. Inform. Theory, vol.34, pp. 35-44, Jan. 1988.
22. S.S. Pradhan and K. Ramchandran, "Distributed source coding: Symmetric rates and applications to sensor networks," Proc. of IEEE DCC, pp. 363-372, Mar. 2000.
23. V. Stanković, A.D. Liveris, Z. Xiong, and C.N. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," Proc. of IEEE DCC, pp. 302-311, Mar. 2004.

24. M. Sartipi and F. Fekri, ”Distributed Source Coding in the Wireless Sensor Networks using LDPC codes: The entire Slepian- Wolf Rate Region,” Proc. of IEEE WCNC, pp. 1939-1944, Mar. 2005.
25. N. Gehrig and P.L. Dragotti, ”Distributed Compression in Camera Sensor Networks”, Proc. of IEEE MMSP, Sept. 2004.
26. S. Li and K. Fukumori, ”Spherical Stereo for the Construction of Immersive VR Environment, ” Proc. of IEEE VR, pp. 217-222, Mar. 2005.
27. C. Geyer and K. Daniilidis, ”Catadioptric projective geometry,” International Journal of Computer Vision, vol. 45(3), pp. 223-243, Dec. 2001.
28. M. Flierl and B. Girod, ” Multiview Video Compression - Exploiting Inter-Image Similarities,” IEEE Signal Proc. Magazine, Special Issue on Multiview Imaging and 3DTV, vol. 24(6), pp. 66-76, Nov. 2007.
29. M. Flierl, A. Mavlankar, and B. Girod, ”Motion and Disparity Compensated Coding for Multiview Video,” IEEE Trans. on Cir. and Sys. for Video Tech., Special Issue on Multiview Video Coding, vol. 17(11), pp. 1474-1484, Nov. 2007,