

DISTRIBUTED COMPRESSION FOR LARGE CAMERA ARRAYS

Xiaoqing Zhu, Anne Aaron, and Bernd Girod

Stanford University
Information Systems Laboratory, Department of Electrical Engineering
350 Serra Mall, Stanford, CA 94305, USA

Invited Paper

ABSTRACT

We address the problem of compression for large camera arrays, and propose a distributed solution based on Wyner-Ziv coding. The proposed scheme allows independent encoding of each view with low-complexity cameras, and performs centralized decoding with side information from additional views. Experimental results are given for two light field data sets. The performance of the proposed scheme is compared with independently coding each view using JPEG2000 and a shape-adaptive JPEG-like coder. The Wyner-Ziv coder yields superior compression performance at low bit-rates. In addition, there is a great reduction in encoder complexity when compared to JPEG2000.

1. INTRODUCTION

Large camera arrays can capture multi-viewpoint images of a scene, which might be used in numerous novel applications ranging from surveillance to movie special effects. In their seminal paper [1], Levoy and Hanrahan suggest the use of light fields, a sampled representation of the light radiating from an object or scene, for image-based rendering. For camera arrays built for such applications, one of the challenges is the enormous size of raw data, typically consisting of hundreds of pictures. Hence, compression is needed.

To exploit the coherence among neighboring views, the images are usually encoded jointly. In large camera arrays, however, cameras typically can only communicate with a central node, but not amongst each other. Since joint coding at the central node requires transmission of all raw images first and excessive memory space to store them temporarily, it is preferable to compress the images directly at each camera, in a distributed fashion. Existing systems either rely on built-in compression capabilities of the capturing devices, thus requiring expensive cameras, or need to add customized circuits to perform some form of standard image compression such as JPEG. With hundreds of cameras involved, the cost of either approach may be prohibitive.

We propose a distributed compression scheme based on a Wyner-Ziv codec, initially designed for intraframe encoding and interframe decoding of motion video [2]. The proposed scheme assumes no communication between the cameras and requires only a very simple, low-complexity encoder at each camera. The burden

of computation is shifted to the centralized decoder, which is assumed to be more sophisticated. In the case of light field compression, the decoder also needs to perform scene geometry estimation, rendering of side information and adaptive rate control.

The remainder of the paper is organized as follows. In Section 2 we review the development of Wyner-Ziv coding from theory to practice, and introduce the structure of a Wyner-Ziv codec based on turbo codes. In Section 3 we provide a system description, explain how to obtain and incorporate the side information at the decoder, and briefly discuss the incorporation of shape adaptation. A simple complexity analysis is given in Section 4 for the proposed scheme versus JPEG2000. In Section 5 we present first results on the compression performance of the proposed coding scheme, in comparison with JPEG2000 and a shape-adaptive JPEG-like coder.

2. WYNER-ZIV CODING

2.1. Prior work

Two results from information theory suggest that a compression system with distributed encoding and centralized decoding can be as efficient as joint encoding and decoding. The Slepian-Wolf theorem states that the achievable rate region for independently encoding two statistically dependent discrete signals is the same as if the two encoders could cooperate [3]. The counterpart of this theorem for lossy source coding is Wyner and Ziv's work on source coding with side information [4]. They derived the rate-distortion bound for the scenario where a side information Y which is related to the source X is not available to the encoder, but can be accessed at the decoder. They also proved that for X and Y jointly Gaussian, the Rate-MSE Distortion performance bound for coding X is the same as if Y is also known at the encoder. We refer to lossy source coding with side information at the decoder as Wyner-Ziv coding.

It has only been recently that practical techniques for Wyner-Ziv coding are studied. Pradhan and Ramchandran presented a practical framework based on syndromes of the codeword cosets [5]. Since then similar concepts have been extended to more advanced channel codes [6]-[9].

Wyner-Ziv coding has also been proposed for applications such as image compression and transmission [10]. Results on applying the Wyner-Ziv codec to motion video coding are also reported in [2].

This material is based upon work supported by the National Science Foundation under Grant No. ECS-0225315 and Grant No. CCR-0310376.

2.2. Wyner-Ziv codec

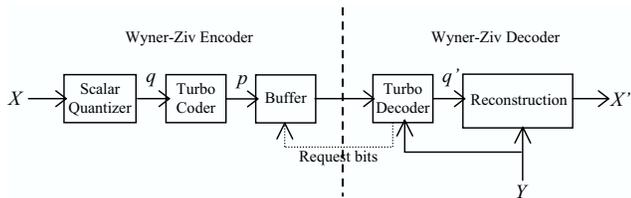


Fig. 1. Wyner-Ziv codec consists of an inner turbo codec and an outer quantization-reconstruction pair.

The Wyner-Ziv encoder and decoder is illustrated in Fig. 1. A similar Wyner-Ziv codec structure was used in [2] for an asymmetric video compression system employing intraframe encoding but interframe decoding.

At the encoder, each pixel within a Wyner-Ziv view X is quantized using a uniform scalar quantizer of 2^M levels. The quantized symbols, q , corresponding to a view are grouped together to form the input block to the turbo encoder. The turbo encoder, composed of two constituent systematic convolutional encoders, generates parity bits p which are stored in an encoder buffer. The buffer transmits a subset of these parity bits to the Wyner-Ziv decoder upon request.

As discussed in more detail in Section 3.2, the decoder has access to some side information Y . The turbo decoder uses the side information Y and the received subset of the parity bits to form the decoded symbol stream q' . If the decoder cannot reliably decode the symbols, it requests additional parity bits from the encoder buffer through feedback. The request and decoding process is repeated until an acceptable probability of symbol error is guaranteed. Using side information, the decoder can request fewer than all M bits for deciding which of the 2^M bins a pixel belongs to, hence achieving compression.

After the receiver decodes q' it calculates a reconstruction for each pixel X' where $X' = E(X|q', Y)$. Assuming that the decoded symbols are correct, the Wyner-Ziv codec limits the distortion of each pixel up to a maximum distortion determined by the quantizer coarseness.

3. WYNER-ZIV CODING

In the following sections we describe the system architecture of distributed coding and centralized decoding for large camera arrays. We then give a detailed account on how to obtain and utilize the side information. For camera arrays depicting an object, the issue of shape-adaptation is also addressed.

3.1. System architecture

The general framework of compression for large camera arrays is shown in Fig. 2. Part of the views are acquired using conventional methods, either in uncompressed form or coded with conventional techniques such as JPEG. The remaining views can be captured using cameras equipped with Wyner-Ziv encoders. For simplicity, cameras with Wyner-Ziv coders are called *Wyner-Ziv cameras*, and those without, *conventional cameras*. Due to the low-complexity of Wyner-Ziv encoding, Wyner-Ziv cameras can be thought of as low-cost sensors. Note that although in the figure

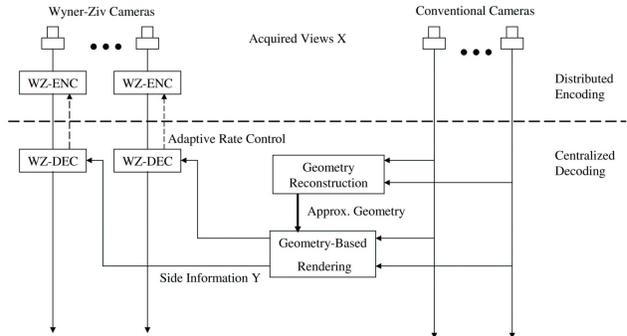


Fig. 2. System architecture of distributed light field compression: views are captured and encoded independently at each camera; they are then decoded jointly. Views from conventional cameras are used to render the side information needed by Wyner-Ziv decoding.

the Wyner-Ziv cameras and the conventional ones are set apart for conceptual clarity, in practice they are interspersed amongst one another to ensure that views from conventional cameras can provide a good estimate for the Wyner-Ziv coded views.

We assume no interconnections amongst the cameras, therefore the views are encoded independently at each camera. The bitstreams are then all transmitted to the centralized decoder. The decoder reconstructs scene geometry from the conventional views, renders an estimate for each Wyner-Ziv view using the geometry, and finally performs decoding with side information. The decoder is also responsible for adaptively requesting bits to achieve certain reconstructed quality, as explained in Section 2.2.

3.2. Side information from rendered views

In order to estimate the Wyner-Ziv encoded views from the neighboring ones already available at the decoder, a rendering procedure analogous to motion-compensated interpolation in video coding is needed. Most rendering algorithms rely on some form of scene geometry [11][12], which can be estimated from multiple camera views using methods described in [13].

Note the analogy between view estimation from neighboring images using geometry-based rendering and frame estimation from adjacent pictures using motion-compensated interpolation. The reconstructed scene geometry provides the disparity information, i.e., the correspondence between different pixel positions in neighboring views, thus serving a similar purpose as that of motion vectors in video. The rendering process is just an interpolation between pixel values from neighboring views corresponding to the same point in 3-D space.

Following similar arguments and observations as in [2], we use a Laplacian model for the residual error between the estimated and acquired pixel values. The parameter α of the Laplacian distribution $f(X - Y) = \frac{\alpha}{2} e^{-\alpha|X - Y|}$ can be estimated at the decoder by fitting the histogram of the difference between the key views and the estimations.

3.3. Shape adaptation

When the cameras are set up to capture the appearance of an object with extraneous background, shape adaptation techniques as

proposed in [14] can be applied to achieve higher compression efficiency. For such cases, the background is usually set to a constant color during the image acquisition stage. Since Wyner-Ziv coding is performed on a pixel-by-pixel basis, we can easily avoid spending bits on the background by a method similar to chroma keying. The encoder can skip pixels of the known background color during its encoding process. Correspondingly, the decoder only needs to decode and estimate symbol error rate for pixels within the object shape, and reconstruct everything outside using the known background color.

Note that to avoid the mismatch of object shape information between the encoder and the decoder, the Wyner-Ziv cameras need to send the object shape information to the decoder. This shape information can be coded using standard techniques such as JBIG [15] and transmitted as overhead.

4. COMPLEXITY ANALYSIS

For large camera arrays, encoder complexity is the major concern. In the following we compare the complexity of the proposed Wyner-Ziv encoder with the emerging standard JPEG2000 [16].

For the Wyner-Ziv coder, encoding of one pixel requires one quantization step, one look-up-table (LUT) operation for the interleaving stage, and two LUTs or feedback shift register operations corresponding to the two constituent convolutional coders. Shape adaptation further reduces the total number of pixels that need to be coded, at the price of more complexity in shape extraction and coding.

JPEG2000, on the other hand, requires multi-level discrete wavelet transform, data partitioning of the coefficients into blocks, and context-based adaptive arithmetic coding. With the typically chosen bi-orthogonal 9/7 wavelet kernel, at least 4 multiplications and 8 additions per pixel are required for one level of 2-D DWT even with the efficient lifting implementation [16]. Generally more than 3 levels of decomposition are needed to fully exploit the spatial correlations between the pixels, resulting in at least 5 multiplications per pixel. More complexity is introduced by the context-based arithmetic coder.

From the simple calculations above, it is obvious that the complexity for Wyner-Ziv encoding is much lower than that of conventional image coding. This allows the image acquisition system to use large arrays of low-cost cameras capable of compression.

5. EXPERIMENTAL RESULTS

We present experimental results on two light field data sets. *Buddha* is a synthetic data set with 280 views and 512×512 pixels in each view. *Garfield* captures a real world object using a hemispherical camera setting containing 8 rows and 32 columns of views, each at a resolution of 384×288 pixels. All experiments are carried out on the luminance component only. Reconstruction quality is measured in terms of *Peak-Signal-to-Noise-Ratio* (PSNR), and bit-rates are expressed as *bits per pixel* (bpp).

For comparison, we also encode each view independently using JPEG2000 and a JPEG-like coder based on the Shape-Adaptive DCT (SA-DCT), as described in [14]. We only compare the rate-PSNR performance and reconstructed image quality for the views captured by Wyner-Ziv cameras, i.e., half of the entire data set, as the other half of images are treated in the same way for all three schemes.

For the synthetic *Buddha* data set, uncompressed conventional views and perfect geometry model are used to render the side information. For *Garfield*, practical limitations are introduced by using estimated geometry and reconstructed images after JPEG2000 compression at 0.25 bpp. We apply shape adaptation to both data sets for the proposed and the SA-DCT coder. There is no need to transmit shape information for *Buddha* since the decoder can derive it from perfect geometry. Whereas for *Garfield*, shape information is coded at 0.0814 bpp using JBIG, and counted toward the total bit-rate as overhead.

The rate-PSNR curves and sample reconstructed images are shown in Fig. 3 for *Buddha* and Fig. 4 for *Garfield*. Due to the help of side information at the decoder, the Wyner-Ziv coder performs significantly better than the other two schemes in the low bit-rate range. For *Buddha*, the performance gain is up to 4 dB in PSNR. For *Garfield* the improvement is around 2 dB over JPEG2000 and about 4 dB over the JPEG-like coder. At higher bit-rates, however, JPEG2000 and the SA-DCT coder tend to be more efficient, whereas reconstruction quality of Wyner-Ziv coding is limited by quantizer coarseness. Also note that due to the overhead of shape coding, the benefit of shape-adaptation is compromised in the case of *Garfield*, therefore the SA-DCT coder performs worse than JPEG2000.

As shown in (b)-(d) of both figures, the relative quality of the reconstructed images reflect the same trend. At low bit rates, JPEG2000 tend to blur out image details and incur ringing effects at object boundaries. The SA-DCT coder preserves the object boundary, but introduces blocking artifacts. The pixel-domain Wyner-Ziv coder, on the other hand, benefits from the rendered side information, and gives sharper details of the images.

We also compare the CPU execution time of encoding both data sets on the same Pentium IV 1.7GHz computer. On average, it takes the Wyner-Ziv coder 23.7 milliseconds(ms) to encode each view of *Buddha* and 5.95 ms per view to encode *Garfield*, whereas JPEG2000 needs 77.4 ms and 38.8 ms to encode each view of the two data sets, respectively. Note that the Kakadu software we use for JPEG2000 is highly optimized, which is not the case for the Wyner-Ziv codec.

6. CONCLUSIONS

We propose a distributed compression scheme for large camera arrays. By applying Wyner-Ziv coding, the proposed scheme makes use of low-cost cameras with low-complexity distributed encoders. Side information is generated from neighboring views via geometry-based rendering. Decoding is performed in a centralized manner.

Experimental results show superior performance of Wyner-Ziv coding over schemes applying independent compression to each image using JPEG2000 and a shape-adaptive JPEG-like coder in low bit-rate regions. There is also a significant reduction in complexity when compared to JPEG2000.

7. REFERENCES

- [1] M. Levoy and P. Hanrahan, "Light field rendering," in *Computer Graphics (Proceedings SIGGRAPH 96)*, August 1996, pp. 31–42.
- [2] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, California, Nov. 2002.

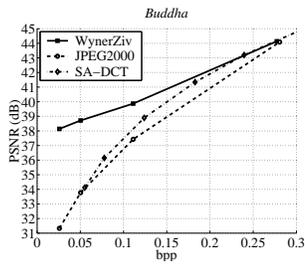


Fig. 3. Performance comparison of Wyner-Ziv coder, JPEG2000 and SA-DCT: *Buddha*: (a) Rate-PSNR curve of the three coders; (b) Wyner-Ziv coder at 0.11 bpp, reconstruction at 39.87 dB; (c) JPEG2000 at 0.11 bpp, reconstruction at 37.43 dB; (d) SA-DCT coder at 0.12 bpp, reconstruction at 38.89 dB.

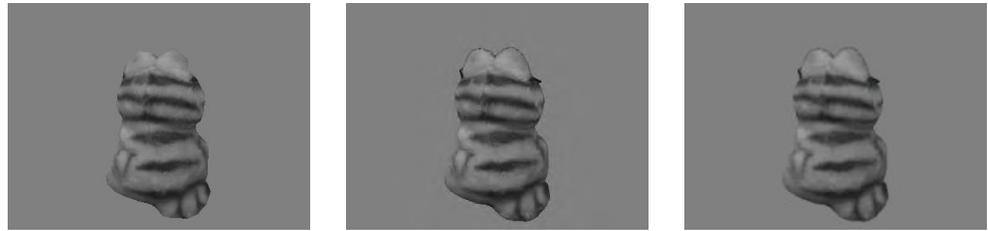
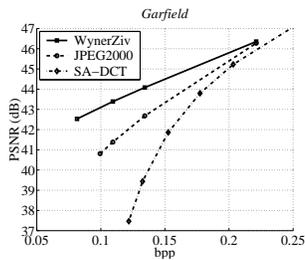


Fig. 4. Performance comparison of Wyner-Ziv coder, JPEG2000 and SA-DCT: *Garfield*: (a) Rate-PSNR curve of the three coders; (b) Wyner-Ziv coder at 0.13 bpp, reconstruction at 44.08 dB; (c) JPEG2000 at 0.13 bpp, reconstruction at 42.68 dB; (d) SA-DCT coder at 0.15 bpp, reconstruction at 41.86 dB.

- [3] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, no. 4, pp. 471–480, July 1973.
- [4] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.
- [5] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Mar. 1999, pp. 158–167.
- [6] J. Garcia-Frias, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, vol. 5, no. 10, pp. 417–419, Oct. 2001.
- [7] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. IEEE Global Communications Conference*, San Antonio, Texas, Nov. 2001, vol. 2, pp. 1400–1404.
- [8] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Apr. 2002, pp. 252–261.
- [9] A. Liveris, Z. Xiong, and C. Georghiadis, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.
- [10] S. S. Pradhan and K. Ramchandran, "Enhancing analog image transmission systems using digital side information: a new wavelet based image coding paradigm," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Mar. 2001, pp. 63–72.
- [11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Computer Graphics (Proceedings SIGGRAPH 96)*, August 1996, pp. 43–54.
- [12] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Computer Graphics (Proceedings SIGGRAPH 01)*, August 2001, pp. 425–432.
- [13] P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 2, pp. 261–277, March 2000.
- [14] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Shape adaptation for light field compression," in *(accepted) IEEE Int. Conf. on Image Processing (ICIP-2003)*, Barcelona, Spain, 2003.
- [15] ITU-T, "ISO/IEC 11544:1993, Progressive Bi-level Image Compression," 1993.
- [16] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, pp. 289–292, Kluwer Academic Publishers, 2002.