TOWARDS A MULTI-TERMINAL VIDEO COMPRESSION ALGORITHM USING EPIPOLAR GEOMETRY

Bi Song, Ozgun Bursalioglu, Amit K. Roy-Chowdhury, Ertem Tuncel

University of California, Riverside, Dept. of Electrical Engineering, Riverside, CA 92521
E-mail: {bsong,ozgun,amitrc,ertem}@ee.ucr.edu

ABSTRACT

We present a novel distributed video coding algorithm based on transform coding of distributed sources and exploiting the geometrical relationships between the location of the sensors. The geometry is used to align the video sequences and distributed quantization of transform coefficients is used to eliminate spatial and inter-sensor redundancy. In contrast with most of the current video compression standards which only exploit spatial and temporal redundancy within each video sequence, we also consider the significant redundancy *between* the sequences. Results demonstrate that our algorithm yields a significant saving in bit rate on the overlapping portion of multiple views.

1. INTRODUCTION

Transmission of video data from multiple sensors over a wireless network requires enormous amount of bandwidth, and could easily overwhelm the system. However, by exploiting the redundancy *between* the video data collected by different cameras, in addition to the inherent temporal and spatial redundancy *within* each video sequence, the required bandwidth can be significantly reduced. Well-established video compression standards, such as MPEG1, MPEG2, MPEG4, H261, and H263, all rely on efficient transform coding of motion-compensated frames, exclusively using the discrete cosine transform (DCT). However, they can only be used in a protocol that encodes the data of each sensor independently. Such methods would exploit spatial and temporal redundancy within each video sequence, but would completely ignore the significant redundancy between the sequences.

In this paper, we develop a novel multiterminal video coding algorithm combining distributed source coding (DSC) and computer vision techniques. This lossy compression scheme takes into account the correlation between the video sensor data, and at the same time keeps the communication between the sensors at a minimum. In broad terms, our scheme relies on alignment of the 2-D video sequences using the epipolar geometry [6] relating the cameras (which could be located arbitrarily in space), followed by standard elimination of temporal redundancy (e.g., via motion compensation), by application of a suitable transform, and finally by quantization of the transform coefficients in a distributed fashion [9]. The epipolar geometry refers to the relationship between the positions of a stereo camera pair, and can be estimated from a pair of stereo images obtained from these cameras. The performance of our algorithm depends, most crucially, on the quality of alignment and the coding efficiency of the distributed quantization scheme. The alignment must result in correspondences between pixels that are maximally correlated, and the distributed coding must optimally exploit this correlation.

It is worth noting that there has recently been significant effort in application of DSC techniques to video data. However, to the best of our knowledge, work on distributed compression in a multi-camera setting using epipolar geometry to reduce inter-camera redundancy has not been studied before in great depth. In what is broadly known as distributed video coding (e.g., [5, 10]), DSC is utilized either for the exploitation of *temporal* correlation in a single video stream, or for better error resilience. A recent method [11] attempts to exploit the redundancy between images available at different sensor nodes by independently encoding the images in low resolution and decoding using superresolution techniques. The high correlation between the low-resolution images, however, is not exploited, and therefore higher coding gains promised by multiterminal source coding theory [1, 7, 8] are not reached. Another recent work [3] developed a distributed image coding technique for a multi-camera setting with several restrictive constraints: cameras are located along a horizontal line, the objects are within a certain known range from the cameras, and the image intensity field is piecewise polynomial. For image-based rendering applications, [13] exhibits a successful algorithm for Wyner-Ziv coding of the light field whereby complexity is shifted from the encoders to the decoder, *but geometrical relationships between camera positions is not taken into account*.

The rest of the paper is organized as follows. Section 2 presents an introduction of transform coding of distributed sources. Section 3 presents an overview of our approach to distributed video coding. In Section 4, some experimental results are presented. Finally, Section 5 gives the conclusion and the future work.

2. TRANSFORM CODING OF DISTRIBUTED SOURCES

The fundamental ingredient of DSC, both in lossless and lossy cases, is *binning* [1, 8], i.e., a many-to-one mapping of the actual data taken from the sources to a limited number of values. Through binning, the correlation between the sources can be exploited without any communication between the sensors.

For two maximally correlated pair of blocks from each view, we use the discrete cosine transform followed by distributed scalar quantization of transform coefficient pairs. Coefficient pairs corresponding to each fixed spatial frequency are encoded independently. Our scalar coding method is provably competitive (in the sense of approaching the rate-distortion bounds) in high bit rates, which is a promising result for the intended (lower bit-rate) applications. Let the shaded region shown in Figure 1 indicate the *support* of a pair of transform coefficients $\tilde{X}$ and $\tilde{Y}$ we need to quantize. It will suffice to design a coding mechanism which encodes the scalars that are inside the support with a small enough distortion, and simply ignore any pair of values falling outside. The encoding must be performed separately, and therefore the cells used for the covering must consist of Cartesian products of individual intervals. The particular assignment in Figure 1 indeed ensures indispensable *unique decodability*, as each pair of codewords pin-

1-4244-0469-X/06/$20.00 ©2006 IEEE

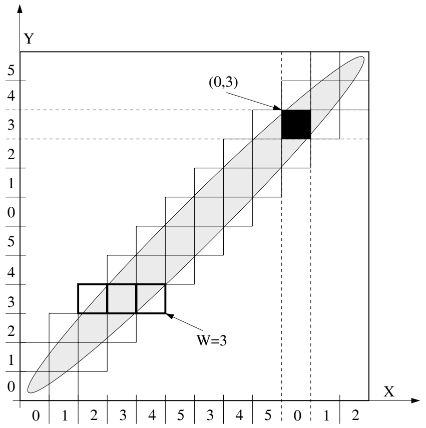

Fig. 1. Proposed coding scheme with $W = 3$, $N_X = 2$, and $N_Y = 2$. As can be seen, 3 cells indicated in bold, suffice to cover the support of the source, indicated as the shaded area, everywhere.

point to a *single* cell that is used in the covering of the support. The example codeword pair shown in the figure, {0,3}, actually corresponds to 4 different cells, but only one of those has a high probability of occurring, and therefore is used for the covering of the support, i.e., as the decoded output. The same statement can be made for all codeword pairs in {0,. . . ,5}×{0,. . . ,5}.

As in [9], we consider a family of codes parameterized by three integers, W, N_X and N_Y. The dynamic ranges of both $\tilde{X}$ and $\tilde{Y}$ are divided into $W \times N_X \times N_Y$ *intervals*, thereby defining a grid on the two dimensional plane. The achieved fixed-length coding rates for the $\tilde{X}$- and $\tilde{Y}$-encoders are $\lceil \log_2 WN_X \rceil$ and $\lceil \log_2 WN_Y \rceil$, respectively. For jointly Gaussian source pairs, we were able to analyze the performance of our scheme rigorously [9], which proved its competitiveness in two aspects: (i) under the uniform high-resolution quantization regime, by separate encoding of $\tilde{X}$ and $\tilde{Y}$, one can achieve the same total distortion one would achieve even if both $\tilde{X}$ and $\tilde{Y}$ were available at a single sensor node [4, Section 8.3], and (ii) under high-resolution assumption, this simple binning technique can attain total rates as close as 3.05 bits to the asymptotical rate-distortion bound characterized in [7].

3. MULTI-TERMINAL VIDEO CODING

Though the issues discussed in the previous sections regarding efficient transform and coding of the data can be applied to any sensor network where nodes observe correlated sequences and are required to transmit their findings to a central receiver, our main focus will be on multi-terminal video compression. The potential gain in multi-terminal is significant due to the inherently high bandwidth of video. Video compression deals with encoding a video sequence after removing the spatial redundancy in each video frame and the temporal redundancy between the frames. Standards such as MPEG1, MPEG2, MPEG4, H261, and H263, outline procedures for achieving this purpose. If we have multiple video sequences from different cameras where there is a significant overlap between the sequences, the above coding standards are inefficient since they do not consider the redundancy in the data at the different sensors. Under this situation, it is necessary to develop distributed video compression schemes that can take advantage of the fact that the data from different sources are correlated. Moreover, this should be done without too much communication between the sensors. Otherwise the savings in bandwidth obtained by considering correlated sources would be offset by the inter-sensor communication.

The theory outlined in the previous section provides an excellent framework to design a distributed lossy video compression scheme. The sensors may be viewing the scene from different viewpoints, but they may have a significant portion of the scene where their field of views (FOVs) overlap. On this overlapping portion, we intend to achieve a very high compression rate using the distributed source coding principles discussed so far. The portion in the frames where there is no correspondence between them will be intra-coded. The geometry between the locations of the sensors will be exploited to understand the correlation between the data at different sensor nodes. We provide below a detailed strategy for developing a distributed video compression strategy for pairs of sensors. Extending it to K sensor nodes will be an issue of future research.

Our scheme relies on obtaining correspondence between the macroblocks (MBs) of the two sensor data at any time instant. Reliable tracking will result in maximally correlated MBs, which, in turn, will lead us to develop an efficient distributed coding scheme. The task of tracking the correspondence of MBs will be achieved by using the motion vectors (MVs) together with the geometrical constraints between the sensors [6]. The MVs can be computed by any scheme, e.g., as in MPEG. The geometrical relationships are expressed through the epipolar constraint. The epipolar constraint states that given a point in one view (say the left image), its corresponding point in the other view lies on the epipolar line. This reduces the search for correspondences to a 1D problem, provided we can compute the epipolar line. This, in turn, requires information about the camera calibration parameters, i.e. the intrinsic parameters of the camera (we will assume that the focal length is the only intrinsic parameter of interest), as well as the extrinsic parameters (i.e., the position and orientation of the camera reference frame with respect to a fixed reference frame in the world). For this paper, we assume stationary cameras, which means that the calibration parameters can be estimated from the images obtained from two or more sensors. Though the present scheme is described for a pair of cameras, it can be generalized to larger sets using the multi-camera constraints [6] or dealing with the cameras pairwise.

3.1. Distributed Motion Estimation (DME) Algorithm

Let us assume that we have two video sensors A and B. Below we provide an overview of our approach for tracking the correspondence between MBs in the two sequences obtained from cameras A and B. We assume, for the purposes of this explanation, that we have calibrated cameras and Camera B knows its position relative to Camera A. We also assume that the two cameras were initially synchronized at time $t = t_1$. The synchronization needs to be done only once at the very start of the transmission and will be available automatically in a recursive manner, by virtue of our algorithm. The main steps of the algorithm, a visual description of which is provided in Figure 2, are listed below.

1. **Problem Definition and Assumptions**: The correspondence between the macroblocks of I_{A1} and I_{B1} is known. The problem is to compute the correspondence between I_{A2} and I_{B2}. The MVs of both pairs of frames $\{I_{A_1}, I_{A_2}\}$ and $\{I_{B_1}, I_{B_2}\}$ are computed separately using an MPEG encoding scheme. That is, both frames I_{A2} and I_{B2} are divided into a uniform grid of macroblocks(MBs) and MVs are computed per macroblock. Let us denote the set of MVs from I_{A2} to I_{A1} by $\{MV(I_{A2})\}$, similarly those from I_{B2} to I_{B1} by $\{MV(I_{B2})\}$. The algorithm, described below, is repeated for each macroblock.

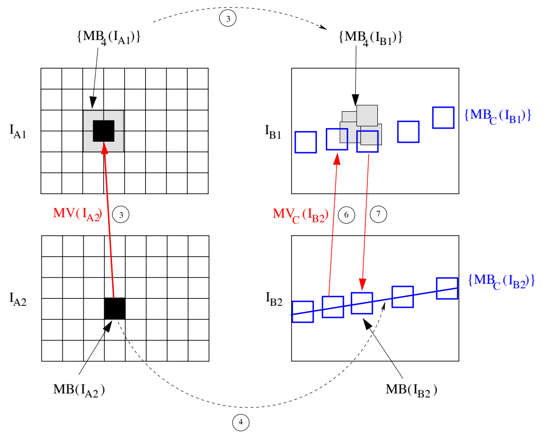

Fig. 2. Pictorial description of the proposed correspondence tracking algorithm. The numbers in circles indicate the steps of the algorithm.

2. **Inter-sensor Communication:** Sensor A transmits $\{MV(I_{A2})\}$ to Sensor B. This constitutes the only communication between the two sensors.
3. For each macroblock $MB(I_{A2})$, motion vector $MV(I_{A2})$ leads to four possible MBs in I_{A1}, denoted by $\{MB_4(I_{A1})\}$. The centers of these four macroblocks are transmitted to Sensor B (as indicated in Step 2), which obtains the corresponding four MBs, $\{MB_4(I_{B1})\}$, using the correspondence between I_{A1} and I_{B1}.
4. Using the epipolar constraint, sensor B computes the epipolar line for each pixel in $MB(I_{A2})$. In practice, we can consider the center pixel of the MB and its corresponding epipolar line. Denote this line by $EP(I_{B2})$.
5. Sampling the epipolar line (possibly non-uniformly), sensor B creates a sequence of MBs, denoted by $\{MB_C(I_{B2})\}$, the center of each being a sample point on the line. Using $\{MV(I_{B2})\}$, it then interpolates to obtain the MVs of this sequence of MBs, denoted by $\{MV_C(I_{B2})\}$.
6. Using $\{MV_C(I_{B2})\}$, sensor B obtains the corresponding sequence of MBs in I_{B1}, denoted by $\{MB_C(I_{B1})\}$.
7. Among the sequence of MBs $\{MB_C(I_{B1})\}$, sensor B chooses the one with the highest amount of intersection with $\{MB_4(I_{B1})\}$. This MB is then traced back to the frame I_{B2}, and the resultant MB, denoted by $MB(I_{B2})$, is declared to correspond to $MB(I_{A2})$.
8. The process is repeated for every MB in I_{A2}. This establishes a correspondence between macroblocks of I_{A2} and I_{B2}. We can now increment the time counter and go back to Step 3.

We will refer to this algorithm as the Distributed Motion Estimation (DME) algorithm.

3.2. Coding Algorithm

After finding corresponding macroblocks $MB(I_{A2})$ and $MB(I_{B2})$, our scheme will proceed as follows. It will (i) compute the motion compensated frames $MC(I_{A2})$ and $MC(I_{B2})$ at the two sensors separately, (ii) compute the residual frames by subtracting $MC(I_{A2})$ from I_{A2} and $MC(I_{B2})$ from I_{B2}, and (iii) apply the DCT separately to each corresponding MB pair $MB(I_{A2})$ and $MB(I_{B2})$. The distributed coding scheme of Section 2 will then be used on the transform coefficients of $MB(I_{A2})$ and $MB(I_{B2})$. Here, steps (ii) and (iii) eliminate the temporal and spatial redundancies, respectively, as usual. However, the novelty in our scheme is the elimination of the redundancy *between* $MB(I_{A2})$ and $MB(I_{B2})$ by using the distributed scalar quantization method proposed in Section 2.

4. EXPERIMENTAL RESULTS

In our experiments we apply the DME algorithm on real imagery, and the results are shown in Figure 3.

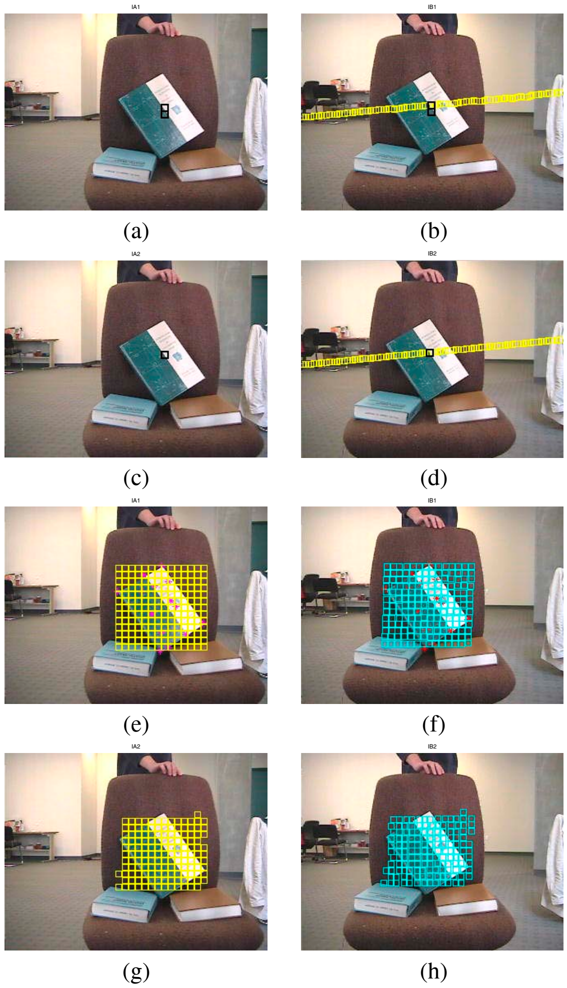

Fig. 3. (a)-(d): Results for one MB. The four images correspond spatially to the images $\{I_{A1}, I_{B1}, I_{A2}, I_{B2}\}$ in Figure 2. (e)-(h): Results for a set of MBs. (e) and (f) represent the initial correspondence between the MBs at Sensors A and B, respectively. (g) and (h) represent the correspondence computed using the DME strategy.

Based on the corresponding macroblocks computed using the DME algorithm, we present in Figure 4 results of our distributed coding algorithm. At this point, we executed our source coding scheme on the actual MBs rather than the residual ones (obtained as the difference between actual and the motion compensated MBs). Further, the run-length coding method that usually follows DCT coefficient quantization in standard coding algorithms is not yet utilized. Instead, we compared fixed-length coding of

DCT coefficients using the conventional and the proposed quantization methods. We used $N_Y = 1$ for all transform coefficients in the proposed quantization method, thereby enjoying no bandwidth saving for frame I_{B_2}. However, for I_{A_2}, which is depicted in Figure 4, we observe either a 2dB PSNR improvement with the same bit rate or 0.125 bits per pixel improvement on the bit rate with about the same PSNR. Note that for standard MPEG-1 video (352×240 and 30 fps), saving 0.125 bpp translates to about 317 Kbps reduction in the overall bandwidth. Compared to 1.2 Mbits, the standard MPEG-1 video rate, this is a significant saving. It is also considerably larger than the cost of transmitting MV's from sensor A to sensor B, which generally does not exceed 100 Kbps.

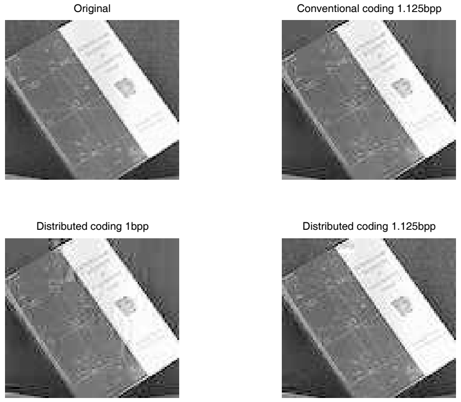

Fig. 4. Point-to-point coding versus distributed transform coding of frame I_{A2}.

We also show in Figure 5 comparison of a particular pair of original MBs with their reconstructed versions. It can be observed that there is some robustness in our distributed coding scheme to small errors in the DME algorithm. Specifically, even though the two MBs are not exactly alike, their significant DCT coefficients more or less are, thanks to the orientation of the edges in the MBs.

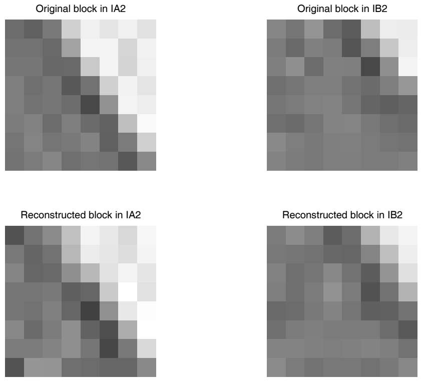

Fig. 5. Comparison of original and reconstructed data for a chosen pair of corresponding MBs according to the DME algorithm.

5. CONCLUSION AND FUTURE WORK

We have designed a novel distributed lossy compression scheme that takes into account the correlation between the video sensor data, and at the same time keeps the communication between the sensors at a minimum. Using epipolar geometry relating the video sensors, the correspondence between the macroblocks of multiple views can be tracked at any time instant, with minimal communication between the sensors. After finding corresponding macroblocks, a suitable transform, and a quantization of the transform coefficients in a distributed fashion are applied to eliminate spatial and inter-sensor redundancy. Using the distributed coding we achieve a better compression rate on the overlapping portion of multiple views.

In general, it is a subject of future work to consider how this scheme can give better performance. First, the coding efficiency can be further increased by using variable-length coding schemes based on non-uniform quantization, as discussed in [2]. We will design such codes, and also investigate the feasibility of schemes such as run-length coding for a block of transform coefficients (as is performed in JPEG coding) for increased coding efficiency in distributed video coding. The accuracy of the video processing algorithms would also be studied so as to reduce errors due to loss of synchronization, misalignment of macroblocks and segmentaion of the common region between two views. We will also consider extension to moving cameras, which will require recalibration of the sensors. Extension to $K > 2$ sensors will be another problem of the future research.

6. REFERENCES

[1] T. Berger, "Multiterminal source encoding," in *The Information Theory Approach to Communications*, G. Longo, Ed., CISM Courses and Lectures 229. Springer, New York, 1978.

[2] O. Bursalioglu and E. Tuncel, "Low-delay distributed source coding: Bounds and perfromance of practical codes," *43rd Allerton Conference*, September 2005.

[3] N. Gehrig and P. L. Dragotti, "DIFFERENT: DIstributed and Fully Flexible image EncodeRs for camEra sensor NeTworks," in *ICIP 2005*.

[4] A. Gersho. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, MA, 1992.

[5] B. Girod, A. Margot, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no 1, pp. 71–83, January 2005.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[7] Y. Oohama, "Gaussian multiterminal source coding," *IEEE Transactions on Information Theory*, vol. 43, no 6, pp. 1912–1923, November 1997.

[8] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no 4, pp. 471-480, July 1973.

[9] E. Tuncel, "Predictive coding of correlated sources," in *IEEE Information Theory Workhsop*, October 2004.

[10] R. Puri and K. Ramchandran, "PRISM: A video coding architecture based on distributed compression principles," submitted to *IEEE Transactions on Image Processing*.

[11] R. Wagner, R. Nowak, R. Baranuik, "Distributed image compression for sensor networks using correspondence analysis and superresolution," in *ICIP 2003*.

[12] A. D. Wyner and J. Ziv, "The rate distortion function for source coding with side information at the receiver," *IEEE Transactions on Information Theory*, vol. 22, no 1, pp. 1–11, January 1976.

[13] X. Zhu, A. Aaron, and B. Girod, "Distributed compression for large camera arrays," in *IEEE Workshop on Statistical Signal Processing*, September 2003.