A MULTI-TERMINAL MODEL-BASED VIDEO COMPRESSION ALGORITHM

B. Song, A. Roy-Chowdhury, E. Tuncel

Department of Electrical Engineering
University of California, Riverside, CA 92521

ABSTRACT

We present a novel 3D model-based distributed video coding algorithm. It is based on independent, model-based tracking of multiple sources and distributed coding of the tracked feature points. The model-based tracking scheme provides correspondence between the overlapping set of features that are visible in the different views. While the motion estimates obtained from the tracking algorithm remove temporal redundancy and the 3D model accounts for removing spatial redundancy, distributed coding is used to eliminate inter-sensor redundancy. Thus, in contrast to 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.

Contribution of the Paper: In this paper, we develop a novel multi-terminal, model-based 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 model-based tracking of individual video sequences captured by cameras (which could be located arbitrarily in space) [15], leading to removal of spatial and temporal redundancies, followed by distributed coding of the tracked feature points [9]. The presence of the 3D model provides correspondence between overlapping feature points in the different views, provided that the tracking of the individual sequences is accurate. The performance of our algorithm depends, most crucially, on the quality of tracking and the coding efficiency of the distributed quantization scheme. The tracking must result in correspondences between pixels that are maximally correlated, and the distributed coding must optimally exploit this correlation.

Relation to Previous Work: It is worth noting that there has recently been significant effort in application of DSC techniques to video data. The novelty of this work lies in the use of computer vision techniques to reduce inter-camera redundancy in a multi-camera setting. 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 multi-terminal 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*. In [16], we presented an algorithm for multi-terminal video compression using epipolar geometry to obtain correspondences between macroblocks (MBs) in two video sequences. This paper shows how that general framework can be extended to the case of model-based compression. The use of a 3D model reduces the inter-sensor exchange since all we have to transmit between cameras is a binary map of the common set of features, and it suffices to transmit this information only when there is an appearance or disappearance of features. Moreover, we avoid the problem of overlapping MBs or gaps between MBs.

The rest of the paper is organized as follows. Section 2 presents the details of the utilized distributed coding scheme. Section 3 outlines the model-based tracking algorithm. Section 4 presents an overview of our approach to distributed video coding. In Section 5, some experimental results are presented. Finally, Section 6 gives the conclusion and the future work.

2. DISTRIBUTED SOURCE CODING

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 ex-

This work was partially funded from Initial Complement funds provided by the Bourns College of Engineering, UCR.

ploited without any communication between the sensors.

For two maximally correlated pair of feature pixels from each view, we use distributed scalar quantization of the pixel values. 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 pixel values X and Y we need to quantize. We devise 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 *unique decodability*, as each pair of codewords pinpoint 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 $\{0,\ldots,5\}\times\{0,\ldots,5\}$.

As in [9], we consider a family of codes parameterized by three integers, W, N_X and N_Y. The dynamic ranges of both X and 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 X- and 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 X and Y, one can achieve the same total distortion one would achieve even if both X and 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]. When used together with non-uniform quantization and variable-length coding, the gap between the rates achieved by this technique and the rate-distortion bound reduces to less than 2 bits [2].

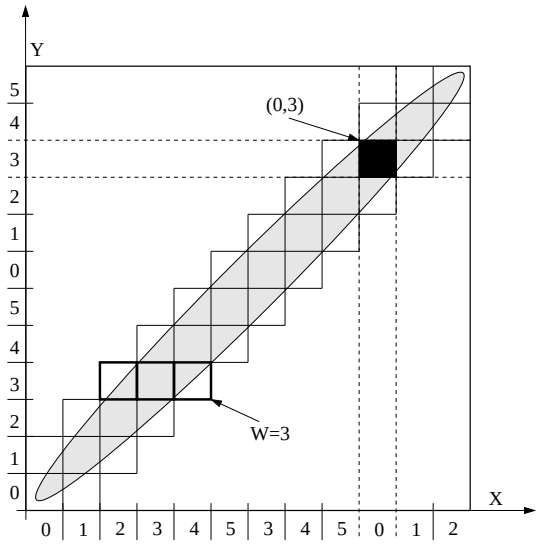

Fig. 1. Proposed coding scheme with $W = 3$, $N_X = 2$, and $N_Y = 2$. As can be seen, 3 cells indicated in bold, suffice to cover the support of the source, indicated as the shaded area, everywhere.

3. MODEL BASED TRACKING

The concept of 3D model-based tracking is well-developed [15]. A texture-mapped model of an object is registered to an image of the object in the first frame. Thereafter, the object is tracked in subsequent images by estimating the 3D translation ($\mathbf{T} \in \Re^3$) and rotation ($\mathbf{R} \in SO(3)$) of the model. This estimation is done by minimizing the reprojection error between the input image and the projection of the 3D model. This is continued for each new incoming frame. Since the current pose is obtained from the prediction of the pose in the previous frame, the problem of drift needs to be taken care of, possibly by re-registering the model to an image. For the purposes of this paper, we used the tracking algorithm described in [14], whose main novelty is handling illumination changes. For the sake of space, we do not go into the details of this algorithm. Any other 3D model-based tracking algorithm would also have served our purpose.

Model-based compression schemes rely on the tracking algorithm to compute the residuals between the input image and the projection of the 3D model using the estimated motion. The 3D model is available at both the encoder and decoder. The encoder transmits the motion estimates and residuals in each frame, while the decoder reconstructs the image at time t by rendering an image using the 3D model and motion estimates and adding the residuals for that frame.

4. DISTRIBUTED MODEL-BASED COMPRESSION ALGORITHM

At the time of registering the 3D object model to the first image in each of the different views, the correspondence of feature points between different views is obtained. This correspondence, which is obtained through the 3D model, remains the same, as the object moves and the features are tracked. The first frame is coded using the distributed scheme described in Section 2 (intra-frame). After that, the estimated motion and residuals are quantized and transmitted, as in any standard 3D model-based compression algorithm [17] (predictive frames). However, due to drift errors, we will need to intra-code after a few predictive frames (as in standard MPEG coders). If re-registration is required, we can use a slight modification of the tracking algorithm, where it starts with the predicted pose and obtains the precise registration [18]. Thus, our scheme relies on distributed coding for intra-frames (I-frames) and separate coding for the predictive frames (P-frames). Since I-frames need the maximum number of bits, the overall savings in transmission rate is substantial. The following is an overview of the algorithm.

Intra-Frame Coding: Register the 3D model to the input image frame in each of the views. Thereafter, the distributed coding scheme of Section 2 is used on the intensity values of each corresponding feature point pair. The correspondence is obtained through the 3D model.

Inter-Frame Coding: Assume that we have encoded and decoded till frame t, and the next frame is I_{t+1}. The encoders for the two views then separately take the following steps.

Step E1: Estimate the motion from I_t to I_{t+1} using the tracking algorithm. Render the image, $\tilde{I}_{t+1}$, using the estimated motion and 3D model.

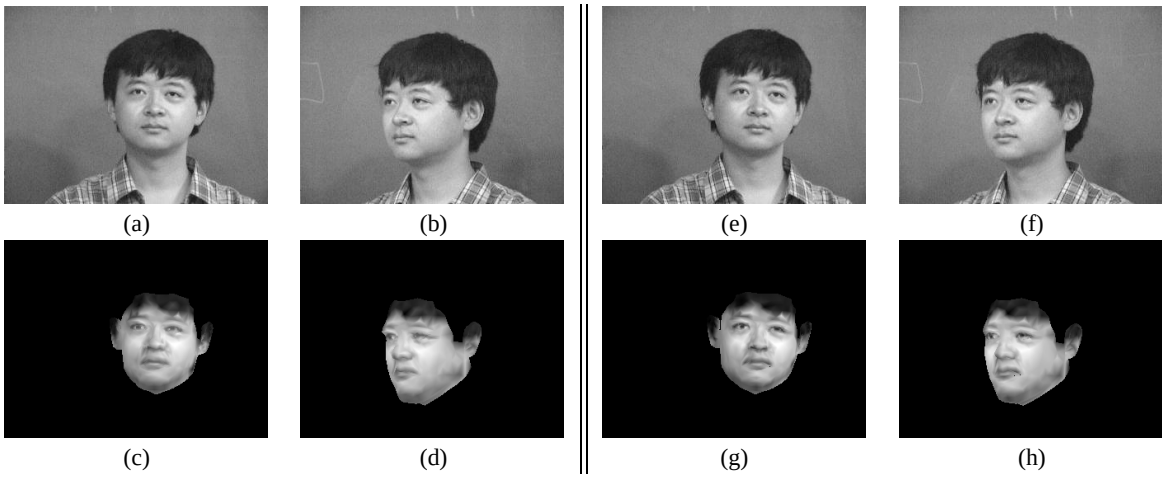

Fig. 2. (a)-(b): Original images of left view and right view respectively for the first frame. (c)-(d): Reconstructed results using the proposed compression scheme. (e)-(f): Original images of left view and right view respectively for the seventeenth frame. (g)-(h): Reconstructed results using the proposed compression scheme.

Step E2: Compute the residual $\delta I_{t+1} = I_{t+1} - \tilde{I}_{t+1}$. If the residual is above a certain threshold, go to Intra-frame coding. Else proceed to next step.

Step E3: Quantize the result from Step E2.

Step E4: Transmit the quantized residuals, along with the quantized rotation and translation vectors.

The following are the actions taken by the decoders.

Step D1: Dequantize the 3D motion estimates and illumination parameters.

Step D2: Dequantize the residuals, and denote it as $\hat{\delta I}_{t+1}$.

Step D3: Using the 3D motion estimates and the 3D model, synthesize an estimate of the rendered image, $\hat{\tilde{I}}_{t+1}$.

Step D4: Obtain the reconstructed image as $\hat{I}_{t+1} = \hat{\tilde{I}}_{t+1} + \hat{\delta I}_{t+1}$.

It should be noted that unlike usual predictive coding schemes such as the DPCM, the in-built decoder at the encoders may not be in synch with the actual decoders. That is because if a pair of corresponding pixel values at the I-frame falls outside of the support set indicated in Figure 1, their reconstruction at the decoder will be different from the reconstruction estimate at the encoder. Even though the P-frame coding is traditional, this possibly out-of-synch initialization of the reconstruction may cause drift. One remedy to this problem is to use a low prediction coefficient ρ, and use $\delta I_{t+1} = I_{t+1} - \rho \tilde{I}_{t+1}$ and $\hat{I}_{t+1} = \rho \hat{\tilde{I}}_{t+1} + \hat{\delta I}_{t+1}$, so that the discrepancy between the built-in decoders and the actual decoders will be quickly forgotten. Analysis of the tradeoff between ρ and the coding efficiency is left as future work.

5. EXPERIMENTAL RESULTS

In our experiments, we applied the distributed model-based compression algorithm on a video sequence of a rotating head. The original images in the first frame (I-frame) in both the sequences and the reconstructed images at the decoder are shown in Figure 2(a-d). The original images in the seventeenth frame (first I-frame followed by sixteenth P-frames) and the reconstruction results are shown in Figure 2(e-h). A binary visibility map of the common set of features is exchanged between the sensors whenever there is an appearance or disappearance of features. The bit-rate for the I-frame is 0.40 bits per pixel (bpp) and the average bit-rate for P-frame is 0.24 bpp.

In Figure 3, we show the effect of choosing W under fixed bit-rate for the I-frame. The horizontal axis represents the average number bits used for each pixel in encoded area of the image (i.e., the face) and the vertical axis is for the PSNR of the reconstructed image compared with the input image. We also compare with the results of the conventional model-based coding of the two sources separately. Employing distributed coding is clearly advantageous over separate coding of the I-frame. More specifically, we observe as high as 3dB gain in PSNR over separate coding. It is also interesting to observe that in low bit rates, $W = 1$ outperforms other values of W. That is because in the low-resolution quantization regime, $W = 1$ already sufficiently covers the support set.

We also show in Figure 4 the distortion of the P-frames, where the number of quantization levels for the residuals is fixed as 8, 16, or 32. In this plot, only the first frame is intra-coded and all subsequent frames are P-frames. The plot shows that around the fifteenth frame, there is a decrease in the achieved PSNR, and thus we need to switch to intra-coding. However, it should be borne in mind that this is the distortion between the decoded image and input image. In a distributed scheme, the in-built decoder within the encoder will not have access to this result. Thus the actual determination of the instance where the P-frame sequence is broken by an I-frame will have to be made on the basis of the residuals in each sequence (i.e., the sum of tracking and quantization errors). We have not analyzed this issue in this paper in detail and obtaining a robust scheme for determining I-frames in a distributed setting is part of our future work.

6. CONCLUSIONS

In this paper, we have presented an algorithm for distributed, model-based video compression. Our scheme relies on 3D model-based

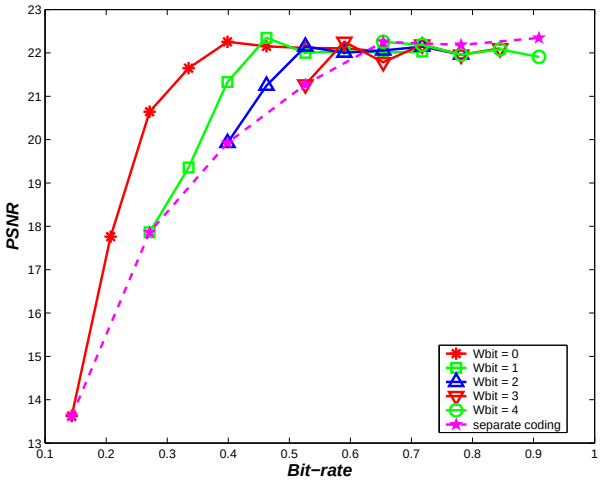

Fig. 3. PSNR of distributed coding of different W values and conventional coding at the same average bit-rates.

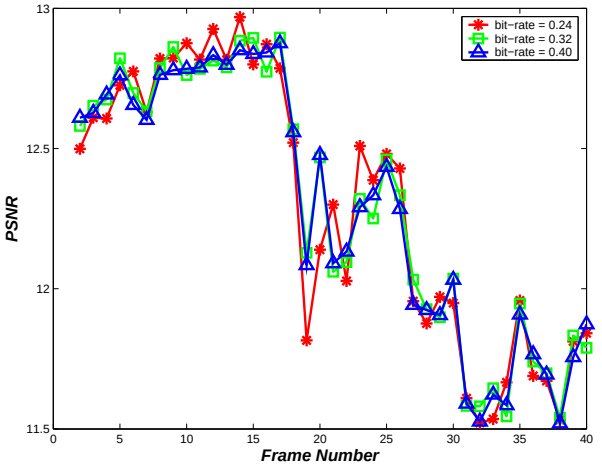

Fig. 4. PSNR for P-frames.

tracking algorithm that operates independently on each of the video sequences. The tracked features points are coded using a combination of distributed compression and predictive coding schemes. I-frames are coded using the distributed scheme, while P-frames are reconstructed from the transmitted motion estimates and image residuals. Since I-frames account for a significant amount of the total bit budget, significant savings in resources is possible using this scheme.

7. 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 performance of practical codes," in *43rd Annual Allerton Conference on Communication, Control, and Computing*, Montecillo, IL, 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 and R. Gray. *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.

[14] A. Roy-Chowdhury, and Y. Xu, "*Integrating Motion and Illumination Models for 3D Tracking*", in Computer Vision for Interactive and Intelligent Environments, Eds. C. Jaynes and B. Collins, IEEE Press, 2005.

[15] V. Lepetit, and P. Fua, "*Monocular Model-Based 3D Tracking of Rigid Objects*", NOW Publishers, 2005.

[16] B. Song, O. Bursalioglu, A. Roy-Chowdhury, E. Tuncel, "Towards A Multi-terminal Video Compression Algorithm Using Epipolar Geometry", *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2006 (Accepted).

[17] T. Ebrahimi and M. Kunt, "Object Based Video Coding", *Handbook of Image and Video Processing*, pp. 585-596, 2000.

[18] Y. Xu and A. Roy-Chowdhury, "Illumination Invariant Facial Pose Estimation and Registration", *IEEE Intl. Conf. on Image Processing*, 2006. (Submitted)