BITSTREAM-BASED CORRELATION DETECTOR FOR MULTI-VIEW DISTRIBUTED VIDEO CODING APPLICATIONS

Charles D. Creusere and *Ivan Mecimore*

Klipsch School of Electrical and Computer Engineering
New Mexico State University
Las Cruces, NM

ABSTRACT

In this paper, we develop a low complexity algorithm for spatial overlap detection and characterization that operates directly on the bitstream of motion-JPEG compressed video. Its low complexity and the fact that it does not require video decoding at the sensor nodes make it well suited to multi-view distributed video coding applications for wireless sensor networks.

Index Terms— Distributed video coding, multiview video coding, bit-domain processing, JPEG-based overlap detection, sensor network coding, morphological filtering

1. INTRODUCTION

While the information theoretic roots of distributed source coding go back to the 1970s with work of Slepian and Wolf and Wyner and Ziff [1], [2], it is only in recent years that practical algorithms have been developed that use the parity bits generated by channel codes to correct for differences between the actual sensed signal and an approximation of that signal derived from side information [3]. The advent of such algorithms combined with a surge in interest in distributed sensor networks has lead to a great deal of recent research in the area and has resulted in an offshoot of this research concept called distributed video compression (DVC) [4], [5].

Our interest here lies in the problem of multi-view distributed video coding [6]-[10]. In this application, one is performing distributed source coding in the classical sense by trying to exploit any correlations that might exist between spatially separated cameras. As an example, consider two video cameras located some small distance apart: clearly, there will be correlations whenever the cameras' fields of view overlap.

The major focus of recent work in this area has been on the problem of synthesizing the side information in the joint decoder [6]-[10]. In every case, it is assumed that the individual encoders are unable to communicate amongst themselves and that the decoder must extract frame correlations without assistance from either encoder. The fundamental difficulties with all of these approaches include poor side information synthesis at the decoder, inefficient parity bit generation for the Wyner-Ziv encoder, and real-time limitations of decoder feedback for rate control. All of these limitations of the currently accepted paradigm have led us to consider an alternative: allow passive communications between video sensor nodes. What we mean by passive is simply to allow nodes to listen to the communications being sent from other nodes to the base station (where joint decoding is performed). Our goal here is still to perform very low complexity video encoding at each node, but to allow nodes to take advantage of what they might 'overhear' from nearby nodes. For example, one node might be able to determine by passively monitoring and analyzing the communications of a nearby node that a portion of its camera's field of view is currently overlapping that of the other node's camera.

The major difficulty with the passive communications paradigm described above is encoder complexity. Specifically, a sensing node mode must spend energy as well as computational processing power to listen to and analyze the signals received from nearby sensors. The process of 'listening' entails, in general, receiving the RF signal, demodulating it, and finally decoding it to reconstruct the video frames. Once decoded, the intercepted frames must then be compared to frames captured locally to determine correlation (e.g., frame overlap). Finally, this correlation must be exploited in the encoding process to create a reduced-rate bitstream describing the local frame.

The focus of our work here is on efficiently determining the overlap between camera fields of view at one of the cameras by analyzing frames in the compressed bitstream domain. This has a number of advantages over working in the pixel domain: (1) the need to decode the passively captured video bitstream is eliminated saving both power and computational bandwidth, (2) far less information needs to be processed since overlap detection is performed in the compressed domain, and (3) the final system should be more robust to small shifts in the fields of view since both the frame overlap calculation and the

Research supported by the ARO, contract # W911NF-06-1-0441

dependent encoding are performed in the same domain (i.e., the bitstream domain). While the motion-JPEG compression used here is not state-of-the-art, it does most certainly satisfies the requirement that the video encoder should have low complexity, and the proposed approach can be extended to I-frame-only MPEG-1 and MPEG-2 as well.

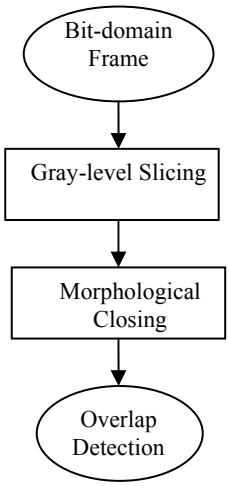

Figure 1: Block diagram of detection overlap detector.

2. BIT-DOMAIN SPATIAL OVERLAP DETECTION

The details of the JPEG image compression algorithm are fully described in [11], but what is important in our application is that it compresses 8x8 pixel blocks of the image in a largely independent manner. Thus, one can view the information in each block as being represented by the number of bits used to encode it. Consequently, the pattern created by the bit counts of the 8x8 pixel blocks that form a region of a video frame provide us information about the spatial composition of that region which can, in theory, be used for spatial matching without needed to decode the video frame.

We henceforth assume that the number of bits used to encode each 8x8 block of the image is extracted directly from a header included with the modified motion JPEG frame. We'll call this the preponderance number (PN) for the block. Because of the way JPEG is encoded, the PN loosely characterizes the frequency content of the block. If the block contains a lot of high frequency energy, then its PN is high; if it does not, then its PN is low. Using this information, we then can match regions in overlapping images using the patterns formed by their preponderance numbers. This is preponderance of bits detection or PBD for short.

Figure 1 shows the block diagram of the proposed PBD scheme. First, gray-level slicing [1] is performed on the PB image i.e. the 2-D array *p(x,y)* containing the preponderance numbers. Experimentally, we have found that slicing around the mean of the entire PB image produces the best result: i.e.,

$$H(x,y)=\begin{cases}1, & \text{if } p(x,y)\geq \text{mean}\,(p(x,y)) \\ 0, & \text{otherwise}\end{cases} \quad (1)$$

After the gray-level slicing, we remove isolated areas of 1s and 0s by using the morphological operation of closing. To implement the closing operator, dilation is first performed using a 3x3 rectangular structuring element followed by erosion with the same structuring element. The purpose of this process is to accentuate changes in neighboring preponderance numbers. Ideally, it will separate the video frame into regions of high and low frequency content. The steps described thus far can be implemented entirely using binary and integer operations with the need to store only zeros and ones.

The detection process applied next attempts to match up related high and low frequency regions in the two frames being evaluated. In developing overlap detection for Fig. 1, we assume that a small amount of side-information is available describing the relative placement of cameras. Specifically, we assume here that the only information available is which side camera 1 is on relative to camera 2—information that can easily be extracted from relatively imprecise non-differential commercial GPS units. This allows us to narrow our overlap search, but it is not fundamentally required for our approach to be effective. Since we know how the cameras are positioned relative to one another, we also know which sides of the images might possibly overlap. Consequently, we start our detection process in the top-left corner of the rightmost camera image. A block size *sqs* is specified, good values for which have been found experimentally to be somewhere between 20 and 30. A block *B* of size *sqs*× *sqs* is then extracted from the top-left corner of the rightmost camera. The mean absolute error (MAE) relative to the PN values of the leftmost camera image is calculated using (2) below and the position with the minimum MAE is determined to be the top-left corner of the overlap region.

$$MAE(x,y)=\frac{1}{sqs^2}\left(\sum_i\sum_j\left|B(i,j)-C(x+i,y+j)\right|\right) \quad (2)$$

3. EXPERIMENTAL RESULTS

This section illustrates the performance of the proposed PBD scheme experimentally. First, we evaluate the number of pixels necessary to accurately identify the overlapping portions of two different frames. The two frames were captured from camera positions 13 centimeters apart and with a 6 degree change in the viewing angle. Ten such pairs of images were produced for this experiment, each translated relative to the same real-world scene. This experiment models a true environment by simulating the different viewing angles for multiple cameras. In each

image frame, the overlapping region is uncontrolled and varies between 40 and 128 pixels. Using a matching window size that is 16% of the total image size (a 35x35 window in the PN domain) an accurate match was made 82% of the time. By increasing the relative window size to 27% (a 45x45 window in the PN domain), we decrease the probability of an incorrect match to only 5%.

Table 1. Percentage of correct detection for the Ballroom sequence.

BALLROOM Camera	Matching Window Size 10	20	30	40
1	0.60	0.86	0.84	0.82
2	0.60	0.92	1.00	0.88
3	0.86	0.98	0.96	0.92
4	0.80	0.96	0.94	0.74
5	0.60	0.86	0.90	0.56
6	0.36	0.80	0.94	0.56
7	0.64	0.96	1.00	0.70

Table 2. Percentage of correct detection for the Vassar sequence.

VASSAR Camera	Matching Window Size 10	20	30	40
1	0.86	0.90	1.00	1.00
2	0.26	0.82	1.00	1.00
3	0.04	0.80	1.00	1.00
4	0.62	0.98	1.00	1.00
5	0.52	0.90	1.00	1.00
6	0.18	0.76	1.00	1.00
7	0.74	1.00	1.00	1.00

Table 3. Percentage of correct detection for the Exit sequence.

EXIT Camera	Matching Window Size 10	20	30	40
1	0.14	0.26	1.00	1.00
2	0.08	1.00	1.00	1.00
3	0.00	1.00	1.00	1.00
4	0.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	0.70	0.96	1.00	1.00
7	0.46	0.24	0.86	0.98

A second experiment examines how the number of pixels used in determining the overlapping region affects the accuracy of the match. This experiment uses the first 50 frames of all three sequences from the MERL library [12] with the images from cameras 1 through 7 being compared to images from camera 0. The matching window size was varied in order to demonstrate how the window size affects accuracy. We declare that an accurate match has occurred whenever the detection scheme calculates the overlap to be within 8 pixels of the true overlap. By using all three sequences, we our exercising the proposed detection algorithm over a diverse input set. The experiment also examines how differing amounts of relative shift affect the detection performance. In the Ballroom sequence, detectable pixel shifts range from 0 to 48; in the case of the exit sequence, the range is a much larger 0 to 208. The results of Table 1 through 3 show that using a window size of 20 or greater produces an accurate detection over 91% of the time. By observing the results in the tables and comparing them to their respective sequences, a few interesting patterns develop. Both the Vassar and Exit sequence have large, low frequency areas with rigid structure. When using a small window size, the algorithm does not perform well. This is because very little high frequency structure is preserved in these small windows over large parts of a frame for these sequences. As we increase the window size, however, the results improve and quickly overtake those of the Ballroom sequence because the rigid structures in these sequences allow for very accurate matching compared to the less-rigid human forms in the Ballroom sequence. Another aspect of the detection scheme becomes apparent when examining the results for the Ballroom sequence. Specifically, we see that the accuracy improves as the window size increases up to 30, and then begins to decrease again at 40. We believe this is due to the addition of more high frequency elements (e.g., people), the positions of which are inconsistent between different camera frames. For all of the MERL sequences, a window size of between 20 and 30 appears to result in the most reliable detection.

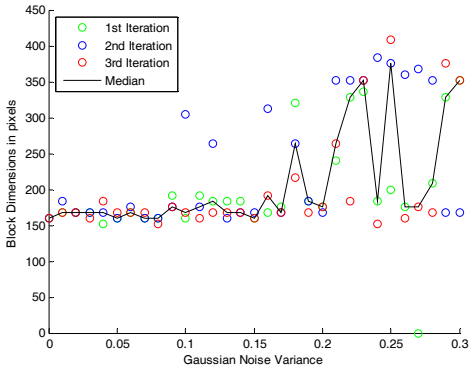

Figure 2: Block dimension for match versus noise variance.

A last experiment was also performed that was designed to characterize the number of pixels necessary to properly identify the overlap between an original image and that image with added Gaussian noise. The minimum number of pixels necessary to make an accurate match for different noise levels is shown in Fig. 2. We see from the figure that the proposed detection scheme is moderately resilient to additive Gaussian noise. In most instances, the PBD algorithm requires less than a 25x25 PN block or about 8.3% of the 800x600 image. As the variance of the Gaussian noise increases, at some point the number of

blocks required for accurate detection must also increase. We believe that the number of pixels required to accurately detect an overlap is related to the useable structure in the preponderance domain. This hypothesis is supported by Fig. 3 which shows the histogram of the block dimensions required for correct detection over a (0, 0.3) range of Gaussian noise variances. By observing the histogram, it is apparent that most detection occurs in clusters of PN block sizes; for example, the first cluster is between 152 and 216, the second between 300 and 408. By comparing this clustering behavior to the spatial domain structure of the image, we can see how that structure affects detection in the bit-domain as the detection window is expanded. As the two images begin to differ from each other more greatly due to higher levels of additive noise, more structure and consequently more PN blocks are needed to accurately detect the overlap. The detection window must then expand until it gathers in enough information to make an accurate decision. Of course, some parts of the image may not provide enough information to counteract the increase in variance; these areas usually have little high frequency content and result in PN block sizes where little overlap detection occurs: i.e., the blank spaces in the histogram of Fig. 3. Interestingly, we have also noticed that the number of blocks used in detection roughly follows the relative entropy for the pixels that map to the regions within the image where the bitstream blocks are being evaluated for overlap. By comparing the histogram to the relative entropy as shown in Fig. 3, we notice that although the magnitudes are very different, the histogram follows the basic shape of the relative entropy. For example, around the 300 and 350 pixel block sizes, the relative entropy has a spike and the histogram seems to follow it. The fact that the histogram follows the relative entropy curve further indicates the utility of bit-domain PBD for extracting information about spatial structure of images.

4. CONCLUSIONS

In this paper we have introduced a novel approach for detecting spatial frame overlap directly from compressed JPEG bitstreams. The preponderance of bits detection scheme that has been proposed here to perform this detection task is being developed to support low power, low complexity distributed video coding and is designed to be simple enough to operate in a remote video sensing node, using information captured passively from other nearby sensors in the network. From the experimental results presented here, we see that the method appears to be effective. Our future research in this project area will focus on using the extracted overlap information to encode frames from one sensor node conditionally with respect to the correlated information sent from the other node.

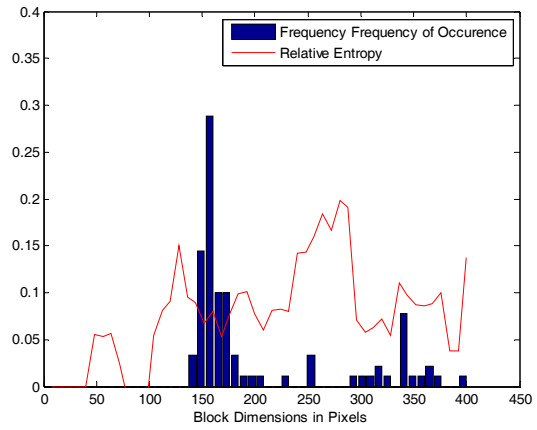

Figure 3: Comparison of block histogram versus relative entropy.

5. REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471-80, July 1973.

[2] A. Wyner and J. Ziv, "The rate distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1-10, Jan. 1976.

[3] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, pp. 51-60, March 2002.

[4] R. Puri, A. Majumdar, P. Ishwar, and K. Ramchandran, "Distributed video coding in wireless sensor networks," *IEEE Signal Processing Magazine*, vol 23, pp. 94-106, 2006.

[5] B. Girod, A. Aaron, S. Rane, and D Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, pp. 71-83, Jan. 2005.

[6] X. Artigas, E. Angeli, and L. Torres, "Side information generation for multiview distributed video coding using a fusion approach," *Proc. 7th Nordic Signal Proc. Symp.*, pp. 250-53, June 2006.

[7] L-W. Kang and C-S. Lu, "Multi-view distributed video coding with low-complexity inter-sensor communication over wireless video sensor networks," *Proc. Int. Conf. on Image Proc* , vol. III, pp. 13-16, Sept. 2007.

[8] I. Tosic and P. Frossard, "Wyner-Ziv coding of multiview omnidirectional images with overcomplete decompositions," *Proc. Int. Conf. on Image Proc.*, vol. III, pp. 17-20, Sept. 2007.

[9] C. Yeo, J. Wang, and K Ramchandran, "View synthesis for robust video compression in wireless camera networks," *Proc. Int. Conf. on Image Proc.*, vol. III, pp. 21-24, Sept. 2007.

[10] Y. Yang, V. Stankovic, W. Zhao, and Z. Xiong, "Multiterminal video coding," *Proc. Int. Conf. on Image Proc.*, vol. III, pp. 25-28, Sept. 2007.

[11] G. Wallace, "The JPEG still picture compression standard," *IEEE Trans. On Consumer Electronics*, vol. 38, pp. xviii-xxxiv, Feb. 1992.

[12] Mitsubishi Electric Research Laboratories, "MERL multiview video sequences," ftp://ftp.merl.com/pub/avetro/ mvc-testseq.