VIEW SYNTHESIS FOR ROBUST DISTRIBUTED VIDEO COMPRESSION IN WIRELESS CAMERA NETWORKS

Chuohao Yeo, Jiajun Wang and Kannan Ramchandran

Dept. of Electrical Engineering and Computer Science
University of California, Berkeley
Email: {zuohao, junewang, kannanr}@eecs.berkeley.edu

ABSTRACT

We propose a method for delivering error-resilient video from wireless camera networks in a distributed fashion over lossy channels. Our scheme is based on distributed source coding that exploits inter-view correlation among cameras with overlapping views. The main focus in this work is on robustness which is imminently needed in a wireless setting. The proposed approach has low encoding complexity, is robust while satisfying tight latency constraints, and requires no inter-camera communication. Our system is built on and is a multi-camera extension of PRISM[1], an earlier proposed single-camera distributed video compression system. Decoder motion search, a key attribute of single-camera PRISM, is extended to the multi-view setting by using estimated scene depth information when it is available. In particular, dense stereo correspondence and view synthesis are utilized to generate side-information. When combined with decoder motion search, our proposed method can be made insensitive to small errors in camera calibration, disparity estimation and view synthesis. In experiments over a simulated wireless channel, the proposed approach achieves up to 2.1 dB gain in PSNR over a system using H.263+ with forward error correction.

Index Terms— Robustness, multi-view, video compression, sensor networks, distributed video compression

1. INTRODUCTION

The practical deployment of wireless camera networks is reliant on a robust infrastructure capable of delivering accurate video streams from the network. The combination of operating in a wireless environment and implementing on sensor mote platforms presents challenges such as bandwidth constraints, lossy channels, low computational capabilities and limited energy supply. This has motivated preliminary work by the authors in the development of methods for compressing and transmitting video from multiple wireless camera sensors in a robust and distributed fashion while minimizing transmission costs and computational complexity [2].

In this work, we explore further in this vein and investigate the use of estimated scene depth information when it is available at the decoder. This, in conjunction with view synthesis, allows us to generate decoder side-information. We also present experimental results demonstrating the superiority of our method over plausible simulcast methods, in which the video stream at each camera is coded and decoded independently without using any interview correlations.

2. RELATED WORK

Prior work on distributed compression of multi-view videos has focused on compression gains. One approach combined distributed source coding with distributed block correspondence tracking [3], but did not exploit any temporal redundancy in experimental studies. Other works have used Wyner-Ziv video coding [4] with a fusion of side-information generated by both temporal and view interpolation [5, 6]. However, the use of feedback is critical in [6], and may not be suitable for certain applications which either demand low latency or do not allow for a feedback channel. Furthermore, their use of an affine scene model may not be sufficiently accurate for complicated scenes.

These works also rely on the assumption of a lossless transmission of video data from individual cameras. As packet drops are to be expected in wireless networks, any data compression and transmission system should be robust in the face of errors. In prior work, we have studied how cameras with overlap provide redundancy that can also be potentially harvested for error resilience [2]. While we work on generalizing the PRISM framework [1, 7], there have been other approaches in the literature that use distributed source coding concepts for error resilient video coding [8, 4].

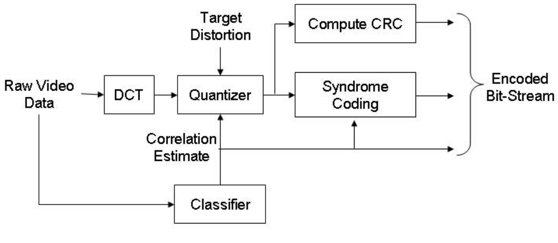
3. APPROACH

In previous work, we extended the PRISM framework, as described by Majumdar et al. [7], to the multi-view setting by using epipolar geometry (which governs the constraint on a single point imaged in two views [9]) to constrain decoder disparity search [2]. In this work, we also made use of dense stereo correspondence and view synthesis when possible to generate side-information for decoder search.

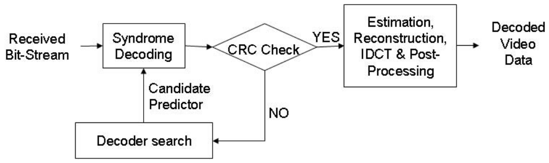
3.1. PRISM overview

The PRISM codec is based on the principles of distributed source coding [1]. Lossy source coding with decoder side-

This work is funded in part by A*STAR.



(a) Encoder block diagram



(b) Decoder block diagram

Fig. 1. System block diagrams.

information is dealt with by the Wyner-Ziv theorem [10], but their results are non-constructive and asymptotic in nature. A practical approach was proposed by Pradhan and Ramchandran [11] and subsequently applied to video coding [1, 12]. The main features of PRISM that are useful in this work are low encoder complexity, use of distributed source coding and decoder motion search. PRISM has demonstrated exceptional robustness to drift while requiring no feedback [7].

Figure 1 shows the block diagrams for the PRISM encoder and decoder. Each video frame is divided into non-overlapping 8x8 blocks. Let $\vec{X}$ denote the current block to be encoded, and let $\vec{Y}$ be $\vec{X}$'s best predictor block in the reference frame. The correlation structure is such that $\vec{X} = \vec{Y} + \vec{Z}$, where $\vec{Z}$ denotes the innovations process. A suitable channel code that is matched to $\vec{Z}$ is used to partition the quantized codeword space of $\vec{X}$, and the syndrome of quantized $\vec{X}$ is transmitted [11]. No motion estimation is performed at the encoder. Instead, a simple classifier based on frame difference is used to estimate the statistics of $\vec{Z}$ and hence determine the rate used to send the syndrome. In theory, the decoder should choose a predictor that is jointly typical with $\vec{X}$ [13]. In practice, a cyclic redundancy check (CRC) on the quantized $\vec{X}$ is also computed and sent. The decoder searches over candidate predictors and attempts to decode using the received syndrome and the candidate predictor as side-information. If the CRC of the decoded sequence checks out, decoding is assumed to be successful.

3.2. Decoder disparity search

Decoder motion search is used in single-camera PRISM to generate candidate predictors for decoding. If predictors in the temporal reference frame are corrupted due to packet drops, but the block to be reconstructed is visible from another camera view, then one way of generating alternative predictors from that view is by sampling blocks along the epipolar line as illustrated in Figure 2. We will refer to this decoding strategy as PRISM-DS (disparity search) [2].

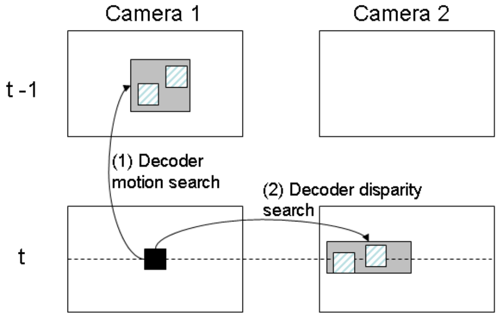


Fig. 2. Decoder disparity search (PRISM-DS). The dark shaded block in frame t of camera 1 is to be decoded. Decoder motion search consists of searching for a predictor in the light shaded area in frame $(t-1)$ of camera 1, while decoder disparity search consist of searching in the light shaded area in frame t of camera 2. The striped blocks are examples of predictor blocks.

3.3. Decoder view synthesis search

Decoder disparity search is a simple way of exploiting inter-camera correlation, since it assumes that a block of pixels in one view can be well predicted by a block of pixels in another view without any further processing. This implicitly assumes that all pixels in the block have the same disparities. If an estimate of the scene geometry is available, then together with view synthesis, we can relax the above assumption. This would at least in theory lead to generating more refined predictors that are better correlated with the original block.

View synthesis using dense depth maps has been used in the past for joint multi-view video compression [14]. In this work, we investigate the use of view synthesis to generate predictors for decoding. As illustrated in Figure 3, if the current frame at camera C is to be decoded, and assuming that the current frames from cameras L and R have been decoded successfully, it is possible to make use of the decoded frames from cameras L and R to synthesize the frame at camera C. To decode a block, the decoder would then sample blocks from a small area around its location in the synthesized frame to use as predictors. As our experimental results will show, the ability to perform decoder motion search is critical in allowing the decoder to be robust to small amounts of calibration, correspondence and interpolation errors.

For specificity, we use a relatively fast and simple stereo correspondence algorithm based on dynamic programming [15] to generate a dense depth map from the current decoded frames at cameras L and R. The estimated depth map is then used in conjunction with a basic view interpolation method [16] to generate an estimate of the current frame at camera C. This will be referred to as PRISM-VS (view synthesis).

4. EXPERIMENTAL RESULTS

In this work, we present two main sets of results. First, we will show that just decoding off the co-located block in the view synthesized image is not sufficient for successful decoding; it is essential to be able to search in a small area to account for inaccurate stereo correspondence and view synthe-

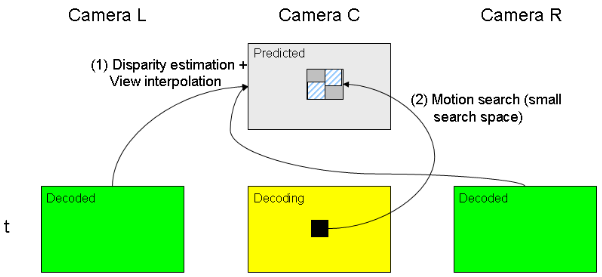

Camera L
Camera C
Camera R
Predicted
(1) Disparity estimation + View interpolation
(2) Motion search (small search space)
Decoded
Decoding
Decoded
t

Fig. 3. Decoder view synthesis search (PRISM-VS). The dark shaded block in frame t of camera C is to be decoded. Decoder view synthesis search consists of first estimating the scene depth map of camera C by using cameras L and R, and subsequently synthesizing an interpolated view at the same location as camera C. The decoder then searches for a predictor in the light shaded area in the predicted view of camera C. The striped blocks are examples of predictor blocks.

sis. Second, we will compare this scheme with the previously proposed decoder disparity search and other simulcast baseline methods.

We used multi-view videos (cropped to 320×240, 30 fps) that were made publicly available by MERL [17], in which eight cameras were placed along a line, at an inter-camera distance of 19.5 cm, with optical axes that are perpendicular to camera displacement. Each camera is assumed to be transmitting over a separate lossy channel. Simulations used packet erasures generated using a two-state channel simulator to capture the bursty nature of lossy wireless channels, with a "good" state packet erasure rate of 0.5% and "bad" state packet erasure rate of 50%. All tests were carried out on a group-of-pictures (GOP) with 25 frames.

4.1. Importance of decoder search in view synthesis

We implemented decoder view synthesis search as described in Section 3.3. We varied the range of the search size (centered at the co-located block) at the decoder. The results shown in Table 1 are for 8% average packet drop rate. As evident, while decoder view synthesis search does help in providing error resilience, we see that its performance saturates only at a search range of about ±2 pixels. This suggests that decoder search is helpful in effectively exploiting side-information generation via dense stereo correspondence and view synthesis. Other distributed video coding schemes [8, 4] code over the entire frame, and hence it would be intractable to try out all combinations of shifts of all blocks from the frame predicted by view synthesis.

4.2. Comparison with simulcast schemes

We compare the performance of our proposed schemes, PRISM-DS and PRISM-VS with the following: (a) single-camera PRISM which uses only decoder motion search (PRISM); (b) Motion JPEG[1] (MJPEG); (c) H.263+ with forward error correction (H.263+FEC); and (d) H.263+ with random intra refresh (H.263+IR). These represent plausible simulcast solutions for multiple cameras.

Table 1. PSNR (dB) with different search ranges (pixels) in decoder view synthesis search. 'None' means no decoder view-synthesis search was performed.

Sequence	±3	±2	±1	0	None
Ballroom	33.07	33.06	33.03	32.80	32.54
Vassar	35.41	35.41	35.41	35.37	35.37

All test systems used the same total rate of 960 Kbps per camera view, with a latency constraint of 1 frame. Each frame is transmitted with 15 packets, with an average packet size of 270 bytes. For H.263+FEC, we used an appropriate fraction of the rate for FEC, implemented with Reed-Solomon codes, such that the quality with no data loss matches that of PRISM. Similarly, we set the intra-refresh rate for H.263+IR such that the quality with no data loss matches that of PRISM.

Figure 4 shows the quality in PSNR of decoded video from camera C. In the "Ballroom" sequence, PRISM-DS and PRISM-VS achieved up to 0.9 dB and 0.4 dB gain in PSNR over PRISM respectively. Compared to H.263+FEC, PRISM-DS and PRISM-VS achieved up to 2.5 dB and 2.1 dB gain in PSNR respectively. In the "Vassar" sequences, both PRISM-DS and PRISM-VS demonstrated modest gains over PRISM.

For visual comparison, Figure 5 shows a portion of the frame from the "Ballroom" sequence after a catastrophic loss event where 60% (reflecting the bursty nature of wireless packet drops) of the previous frame's packets were dropped. Both PRISM-DS and PRISM-VS produced more visually pleasing reconstruction than the other simulcast schemes.

5. CONCLUSION

In deploying wireless camera networks, it is important to design transmission systems that take into the account the lossy nature of wireless communications. We have presented a distributed video compression scheme for wireless camera networks that is not only robust to channel loss, but has low complexity encoders that are highly suitable for implementation on sensor mote platforms. Our experiments demonstrated two results. First, under the PRISM framework, decoder view synthesis generates predictors that matches the performance of predictors generated by decoder disparity search. Second, multi-view video coding systems should be robust to errors in calibration, correspondence and view synthesis to fully reap the benefits of redundancy in overlaps among cameras.

Overall, PRISM-DS seems to perform a little better than PRISM-VS in terms of robustness. This seems to suggest that sophisticated computer vision and computer graphics techniques need not be necessary for distributed multi-view video coding (as was the case in motion compensation for traditional hybrid video codecs).

[1] Simulated by coding all frames as I-frames with a H.263+ encoder. We used a free version of H.263+ obtained from University of British Columbia for our simulations.

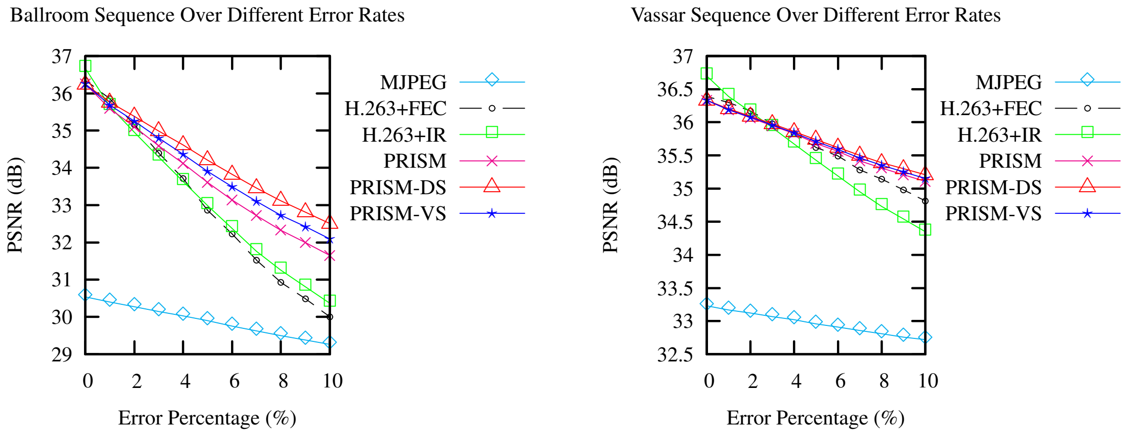
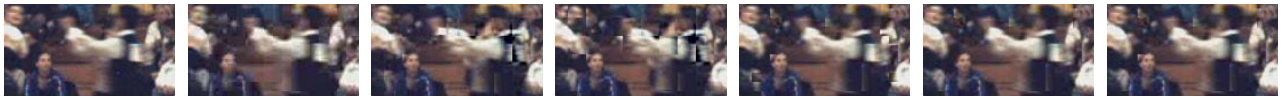


Fig. 4. System performance over different error rates.



(a) Original (b) MJPEG (c) H.263+FEC (d) H.263+IR (e) PRISM (f) PRISM-DS (g) PRISM-VS

Fig. 5. Visual results of "Ballroom" sequence at 8% average packet outage. Note the obvious blocking artifacts in MJPEG, and the obvious signs of drift in both H.263+FEC and H.263+IR. PRISM-DS and PRISM-VS produced reconstructions that are most visually pleasing.

In future work, we would like to explore "smarter" encoders that are able to estimate inter-camera correlation based on intra-camera properties such as edge strength. This would allow us to gain higher compression efficiency. The regime of low frame rate video promises to be an interesting area of research, since inter-camera correlation could possibly dominate intra-camera temporal correlation.

6. REFERENCES

[1] Rohit Puri and Kannan Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conference on Communication, Control and Computing*, 2002.

[2] Chuohao Yeo and Kannan Ramchandran, "Robust distributed multi-view video compression for wireless camera networks," in *Proc. SPIE Visual Communications and Image Processing*, Jan 2007.

[3] Bi Song, Ozgun Bursalioglu, Amit K Roy-Chowdhury, and Ertem Tuncel, "Towards a multi-terminal video compression algorithm using epipolar geometry," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.

[4] B Girod, A M Aaron, S Rane, and D Rebollo-Monedero, "Distributed video coding," *Proc. of the IEEE*, vol. 93, no. 1, pp. 71–83, Jan 2005.

[5] Xun Guo, Yan Lu, Feng Wu, Wen Gao, and Shipeng Li, "Distributed multi-view video coding," in *Proc. SPIE Visual Communications and Image Processing*, Jan 2006.

[6] Mourad Ouaret, Frederic Dufaux, and Touradj Ebrahimi, "Fusion-based multiview distributed video coding," in *Proc. 4th ACM International Workshop on Video Surveillance and Sensor Networks*, Oct 2006.

[7] Abhik Majumdar, Jim Chou, and Kannan Ramchandran, "Robust distributed video compression based on multilevel coset codes," in *Proc. Asilomar Conference on Signals, Systems and Computers*, 2003, vol. 1, pp. 845–849.

[8] A Sehgal, A Jagmohan, and N Ahuja, "Wyner-Ziv coding of video: an error-resilient compression framework," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 249–258, Apr 2004.

[9] Richart Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[10] A D Wyner and J Ziv, "The rate distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, Jan 1976.

[11] S S Pradhan and K Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, Mar 2003.

[12] Anne Aaron, Rui Zhang, and Bernd Girod, "Wyner-Ziv coding of motion video," *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, vol. 1, 2002.

[13] Prakash Ishwar, V M Prabhakaran, and Kannan Ramchandran, "Towards a theory for video coding using distributed compression principles," in *Proc. IEEE International Conference on Image Processing*, Sep 2003.

[14] E Martinian, A Behrens, J Xin, and A Vetro, "View synthesis for multiview video compression," in *Picture Coding Symposium*, Apr 2006.

[15] Sven Forstmann, Yutaka Kanou, Jun Ohya, Sven Thuering, and Alfred Schmitt, "Real-time stereo by using dynamic programming," *Proc. of CVPR Workshop on Real-time 3D Sensors and Their Use*, 2004.

[16] Shenchang Eric Chen and Lance Williams, "View interpolation for image synthesis," *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 279–288, 1993.

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