HOMOGRAPHY BASED DISTRIBUTED VIDEO CODING FOR A NETWORK OF CAMERAS

Ashok Veeraraghavan[†]*, Mahesh Ramachandran*[†] *and Manohar Mareboyana*[††]

[†]Center for Automation Research, UMIACS
University of Maryland, College Park, MD, USA
{vashok,maheshr,rama}@cfar.umd.edu

[††]Department of Computer Science
Bowie State University, Bowie, MD, 20715
manohar@cs.bowiestate.edu

ABSTRACT

Networks of multiple video cameras are being deployed in several scenarios like surveillance, traffic enforcement, human motion analysis and sports telecast. The unmanageable size of the raw video sequences necessitates the use of compression schemes to efficiently encode these videos. Traditional video compression schemes account for spatial and temporal redundancy in a video sequence. In this paper, we present a compression technique that also leverages the information redundancy between video sequences across different cameras with overlapping fields of view. An algorithm based on homography for efficient compression of multiple video sequences is presented that performs significantly better than current schemes. We also derive an efficient distributed version of the algorithm that can be implemented on a large scale network of cameras. This distributed algorithm minimizes both the communication costs and the compression costs simultaneously.

Index Terms— Image Coding, Compression, Camera Networks

1. INTRODUCTION

Large scale networks of cameras are becoming ubiquitous in outdoor surveillance, traffic monitoring, and several other applications. In order to manage and store the voluminous amount of data generated by these video sensors, effective schemes for compression of these videos are needed. Video coding schemes like the MPEG1, MPEG2, MPEG4, H261, and H263 target the intra-sensor redundancy in videos. They exploit spatial redundancy within an image and the temporal redundancy in a video stream by performing transform coding using the discrete orthogonal transforms and motion compensation respectively. In a multi-camera setting, there is also a significant amount of redundancy across overlapping camera views. Traditional video coding schemes do not exploit this inter-sensor redundancy. In this paper, we present a simple, yet powerful technique for video compression that exploits the redundancy present within as well as across video sequences.

We assume multiple cameras with overlapping fields of view, observing a scene containing one or many planes. For instance, one plane may be the dominant ground plane, and there may be other planes defined by buildings or other man-made objects. We utilize inter-sensor redundancy for compression by making use of the homography relationship induced by a plane between images from multiple views. The homography is the only information that we need to perform inter and intra sensor coding. We do not need any other information like intrinsic and extrinsic calibration etc. We experimentally show that in a variety of settings the new multi-video coding technique results in significant savings compared to traditional intra-sensor coding techniques like MPEG. In a sensor network setting, it also becomes important to minimize the required communication bandwidth between the various sensor nodes. We also develop a completely distributed version of the multi-video coding technique that involves minimal information exchange among the sensor nodes during the compression process. This is achieved by a careful selection of the model that relates the observed intensities at corresponding locations on different cameras. We show both quantitatively and empirically that the distributed version of the algorithm results in significant bandwidth savings.

THIS WORK WAS SUPPORTED BY NSF ITR GRANT 0325119

1.1. Prior Work

Existing video compression standards like MPEG1, MPEG2, MPEG4, H261, and H263 do not account for the redundancy across camera views. In the presence of multiple video streams, they act on the each of the individual videos independently. Recently, there have been some efforts to tackle this issue using techniques from distributed source coding(DSC) [1][2]. In [3] and [4] distributed source coding techniques are used to increase error resilience and to move the complexity of the code from the encoder to the decoder. Zhu et. al. [5] developed an algorithm for Wyner-Ziv encoding of light fields in multiple camera settings. But these methods do not exploit the inter-sensor redundancy resulting from the overlapping fields of view of multiple cameras. Wagner et. al. [6] present a method for simultaneous compression of multiple video streams by registering them at a central node and then performing joint coding for the overlapping areas. In another similar approach [7], the parameters of a 3D model are recovered by employing a model-based tracking algorithm to provide registration across camera views. Gehrig and Dragotti [8] developed a distributed source coding scheme for multi-camera images under several restrictive conditions like cameras located on a horizontal line and piecewise polynomial intensity fields. The approach closest to our work is [9], where the epipolar geometry between cameras is used to perform joint source coding. Nevertheless, our approach differs in significant ways. Firstly, [9] require the knowledge of camera locations and calibration information to compute epipolar geometry whereas we do not require complete calibration. We only require the homographies induced by scene planes between images from different views. Moreover, inspired by the results in [2], we describe a distributed version of the algorithm leading to significant communication savings in sensor networks.

2. MOTIVATION AND MULTI-VIDEO CODING

Consider a typical surveillence scenario as shown in Figure 1, where two camera views are surveying a common area. There is a dominant plane, and a several static and moving objects on it. Inspite of significant variations between images from the two views due to pose and illumination effects, the images captured by these cameras are

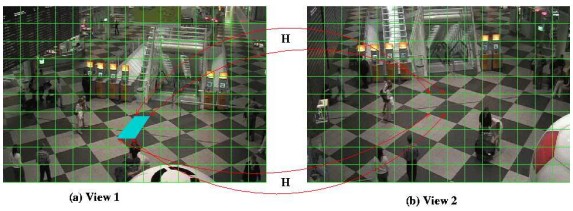


Fig. 1. Two camera views of a typical airport scene. The two images are related, and joint coding will result in compression.

related. In general, the relationship between corresponding pixels in images from the two camera views is expressed by the epipolar constraint. But the epipolar geometry requires knowledge of the exact relative position of the two cameras. Moreover, the epipolar constraint restricts an imaged point in one camera view to lie on a line in the other camera view. The memory savings obtained by using the epipolar constraint is therefore, at best, modest. We note that typical scenes are dominated by the presence of several planar objects. For all points belonging to a particular scene plane, the corresponding pixels in two different camera views are related by a homography. Given the image from one camera, the image of a plane as seen by any of the other cameras can be accurately reconstructed using the homography induced by the plane.

2.1. Homography in multi-video coding

Let us assume that there is one dominant plane in the scene being monitored by the set of cameras. In the scene shown in Fig.1, this is the ground plane. The relationship between corresponding pixels x_1 and x_2 in the two images is given by $x_2 \approx H_{12}x_1$, where H_{12} is a 3×3 homography matrix relating the corresponding points. Instead of coding and storing the image intensities for the images from all the views separately, we could obtain significant coding efficiency in the overlapping field of view by storing the image from one view and the homography matrix relating the two views. But the image intensities of corresponding points might not be identical because of different settings of internal camera parameters like brightness compensation, automatic gain adjustment etc. To model these differences, we write the relation between the intensities of the overlapping portions of two images as:

$$I_2(H_{12}x_1) = f(I_1(x_1), a_{12}) \tag{1}$$

where f is a function modeling the change in gain and brightness, and $I(x)$ denotes the observed intensity at location x. The parameters a_{12} relate the gain and brightness parameters of the two cameras. Therefore, we store only one set of images for a certain viewpoint, the homography between this view and all other views, and a set of camera gain parameters a_{12}. Using this, it is possible to reconstruct the images from all the viewpoints.

2.2. Modeling the Gain and Brightness

The choice of the functional form for f in (1) does not significantly affect the coding/compression efficiency. Nevertheless, certain choices of functions are more preferable than others because they allow efficient network based distributed algorithms to be implemented. In this work, f is chosen to be an affine function of the intensity values at the corresponding pixels with two parameters $a_{12} = \{c_{12}, d_{12}\}$.

$$I_2(H_{12}x_1) = c_{12}I_1(x_1) + d_{12} \tag{2}$$

This choice of functional form accounts for the gain and white balance between the two different cameras. With this functional form, we later derive a distributed algorithm to jointly code the multiple video streams.

2.3. Motion Compensation and Transform Coding

Traditional single stream video coding schemes like MPEG use the paradigm of motion compensation followed by transform coding using DCT in order to exploit the temporal and spatial redundancy and achieve compression. In our scenario, we wish to target the redundancy across video streams in addition to the the redundancies within a sequence. Therefore, we borrow concepts from single stream compression and augment them with the homography based correspondence in order to do better rate-distortion performance than compressing different video streams independently.

Similar to MPEG, we divide each image into 8×8 blocks and compute the $2D$ DCT of these blocks. We quantize these DCT coefficients for each block and then encode them. We then perform run-length coding to leverage the presence of many zero-values in the high frequency DCT coefficients.

2.4. Multi-video coding : The details

In order to account for the redundancy across camera views, the homography-based multi-view coding is implemented, in a block-wise fashion. We describe the algorithm assuming two views. The case for multiple views is easily generalized. The first video sequence is MPEG coded using intra-coded (I) frames, and inter-coded (P and B) frames accounting for motion compensation. The second video sequence is divided into 8×8 blocks. If a block can be adequately reconstructed using the corresponding frame in the first view and the homography, then we only store the gain or white balance (a_{12}), rather than storing the DCT coeficients. Most video cameras have automatic gain and brightness adjustments that associate different gain values to different regions in the image, depending on the observed intensities. To compensate for this spatially varying gain adjustment, we can store the gain-parameters for each block separately rather than storing one set of common values for the entire image. In this setting, using 8 bits for the gain parameters, the entire block can be coded with just 16 bits, i.e. approximately 0.25 bits per pixel, while maintaining the same quality of the observed image as before. When there are occlusions, or when objects are not confined to any of the scene planes, or when there are specularities observed in one view that are not observed in the other view, the homography based reconstruction will result in a large reconstruction error. In all such cases, the corresponding blocks are individually coded using motion compensation followed by transform coding, quantization and run-length coding. This implies that when there is minimal or no overlap between two videos, the compression efficiency of the *Homography based Multi-Video Coding* (HMVC) algorithm is just as good as traditional MPEG encoding of the individual videos.

In the presence of multiple scene planes, we may improve the coding efficiency by using different homographies for different regions of the images. Using the homography relations induced by all the scene planes increases the number of blocks that are encoded using the homography and gain parameters. Additional overhead bits are needed for each block indicating which of the homographies was used for compression. In practice, the overhead is quite small since most scenes have around $3 - 4$ planes which induce homographies between the views. The compression gain obtained by coding additional blocks using multiple homographies is significantly larger than this overhead.

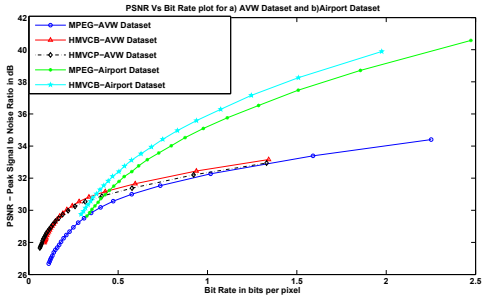


Fig. 2. Plot of Peak Signal to Noise Ratio (in dB) Vs Bit Rate of Encoding in Bits per Pixel per Frame for a) AVWilliams dataset b) Airport Dataset.

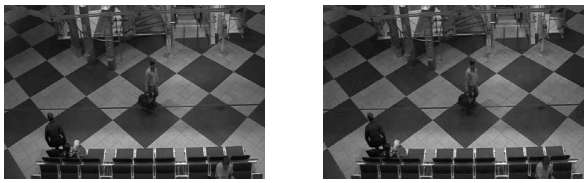
3. EXPERIMENTS ON MULTI-VIDEO CODING

We evaluate the performance of the multi-video coding scheme on two datasets with varying characteristics. Dataset A (A. V. Williams Dataset) contains synchronized video sequences from two cameras observing an outdoor scene with human and vehicular traffic. This dataset exhibits variations in scene brightness, complex motion of targets and parallax due to large vehicles. The second dataset (airport dataset) contains three cameras surveying an airport scene with many moving objects (people) in the scene.

We evaluate the performance by comparing the PSNR versus bit-rate plots with the baseline MPEG coding of the video streams. We implement two versions of our algorithm - one with the affine gain parameters constant all over the image (HMVC-P), and the other with these parameters different for each block (HMVC-B). The first version (HMVC-P) provides us with higher compression efficiency while it introduces some artifacts when the camera gain is different in various image regions. The performance is expected to be similar in scenes with normal dynamic range. But for scenes with high dynamic range, HMVC-B is likely to perform better. Figure 2 shows the peak signal to noise ratio of the various compression algorithms on the AVWilliams datasets. At very low bit rates the HMVC-P algorithm outperforms the other algorithms. At moderate and high bitrates the HMVC-B algorithm performs best. There is a clear increase in performance of HMVC compared to traditional MPEG because of targeting inter-sensor redundancy between video streams. The PSNR versus bit rate plot for the airport sequence is also shown in Fig. 2. For this sequence, the ground plane had a wide variation in texture (checkerboard pattern), therefore we could not obtain estimates of gain-parameters that were consistent across all regions of the image. This suggests that it is important in some scenes to store the gain parameters in a blockwise fashion. Sample output image reconstructed from the compressed data are shown in Figure 3. In this scene, there was significant parallax from the plane because of objects and moving targets. We see that even at very low bit-rates such as 0.5 bits per pixel, the reconstruction using HMVC is good over all regions of the image.

4. DISTRIBUTED ALGORITHM

Consider a sensor network scenario. Each node X_i is a video sensor and is monitoring a scene. There is significant overlap between the fields of view of the various video sensors. Node S is the sink node and is the central facility where all the image sequences are collected and stored for further processing. If we use the HMVC algorithm at the central node S for compression, then we would obtain tremendous memory savings at the central sink node but the required bandwidth of each of the links between the video sensors and the sink node would still be $H(X_i)$ where H denotes the entropy. In practice, the required bandwidth of each of these links would have to be at least as large as the bit-rate of traditional MPEG compression. For a general linear network where encoding is performed only at the sink node, the required capacity of the links is shown in Figure 4(b). The Slepian-Wolf Theorem [1] states that two correlated sources can be compressed to the same extent irrespective of whether the sources communicate or not, as long as decompression takes place at a common receiver. This implies the existence of some distributed coding scheme that can potentially be implemented within the bandwidth constraints shown in Figure 4(d). But practical designs for Slepian-Wolf video codes have been very hard to come by.



(a) Original image (b) HMVCB reconstructed image

Fig. 3. Original and reconstructed images (at 0.5 bits per pixel) for airport scene.

4.1. Correlated Data Gathering

Distributed compression schemes for correlated data have been studied in detail in [2]. They address the problem of joint rate allocation and optimization and show that when the transfer matrix is arbitary, the problem of finding the optimal transmission structure is NP complete while this task can be efficiently achieved when the flow cost is separable. We leverage the results and ideas presented in [2] to derive efficient distributed algorithms for the problem of distributed video coding. In particular, we describe two different distributed coding architectures - one based on explicit communication and the other based on approximation to the distributed version of the HMVC (DHMVC). Most of the results shown here are motivated using either the linear network or the star network, for which explicit bounds and performance guarantees have been obtained in [2]. Nevertheless, the ideas presented here can easily be extended to networks of more general topology by approximating them into a sum of either star or linear networks of appropriate form.

4.2. Explicit Communication

Consider the linear network shown in Fig. 4(c). The nodes named $N_1, N_2, ..N_M$ are each connected to video cameras and need to send the data over to the central processing node S. Denote the video source at node N_i by X_i and its entropy by $H(X_i)$. In the absence of any distributed scheme, the link L_{M-1} between N_M and N_{M-1} needs to be able to accomodate a bandwidth of $H(X_M)$. If the required capacity of each link L_i is denoted by $C(L_i)$, then in the absence of any coding the required capacity of the links is given by

$$C^{NoCoding}(L_i) = \sum_{j=i+1}^{M} H(X_j)$$

In the explicit communication based scheme for multi video coding, the node N_M transmits its entire video stream to the node N_{M-1} via link L_{M-1}. Therefore the required capacity of link

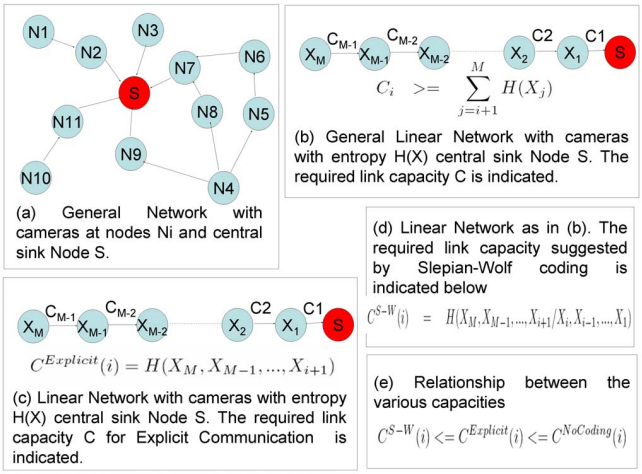


Fig. 4. Network of cameras and the required link capacities for the links for each of the schemes discussed.

L_{M-1} would remain i.e. $C(L_{M-1}) = H(X_M)$. But each of the subsequent nodes, perform multi video coding before transmitting the combined video streams further to the next node. For example node N_{M-1} would jointly code X_M and X_{M-1} using the HMVC algorithm and then would transmit the jointly encoded stream further down to the next node. Therefore, the required capacity of link L_{M-2}, given by $C(L_{M-2})$ would be $H(X_M, X_{M-1})$. Since, the video streams overlap we know that $H(X_M, X_{M-1}) <= H(X_M) + H(X_{M-1})$. In general, the required capacity of any link L_i can be given by

$$C^{Explicit}(L_i) = H(X_M, X_{M-1}, ..., X_{i+1}) <= C^{NoCoding}(L_i). \quad (3)$$

This would turn out to be a significant communication savings for most networks as shown in Figure 4(c).

4.3. Approximate Slepian-Wolf coding for HMVC

Even though the explicit communication based scheme for multi video coding results in significant bandwidth savings for general networks, it still does not reach the limits suggested by Slepian-Wolf theorem [1]. The theorem states that two correlated sources can be compressed to the same extent irrespective of whether the sources communicate or not, as long as decompression takes place at a common receiver. This implies the existence of some distributed coding architecture that can potentially achieve the same encoding gains as that of the proposed HMVC algorithm but with the bandwidth requirements given by Slepian-Wolf as follows:

$$\begin{aligned} C^{S-W}(i) &= H(X_M, X_{M-1}, ..., X_{i+1}/X_i, X_{i-1}, ..., X_1) \\ &<= C^{Explicit}(L_i). \end{aligned} \quad (4)$$

The predicted capacity of the required links are related as,

$$C^{S-W}(i) <= C^{Explicit}(i) <= C^{NoCoding}(i) \quad (5)$$

It is fair to assume that the set of blocks within each image that are H-coded changes very slowly in natural scenarios. We do explicit communication between nodes after every n frames, and find out the image blocks that can be H-coded. Then we H-code this same set of blocks for the next $n-1$ frames. There may be some errors in subsequent frames while doing this, but this results in significant bandwidth savings for a small compromise in reconstruction error. For the airport sequence, we find out the set of all subblocks in the second view that can be coded using the known homography (assuming explicit communication). Then we find out how many of these subblocks are common with those estimated for the first frame. This gives the percentage of the subblocks on which we make correct decision in the case of the approximate distributed scheme described above. The results are shown in table (1). Around 93% of the set of subblocks that are supposed to be H-coded in a particular frame are common with the same set from the first frame.

Frm. no.	2	4	6	8	9
perc.	95.11	94.07	92.79	92.67	92.12

Table 1. Percentage of H-coded subblocks that are common between a certain frame and the first frame

5. CONCLUSIONS AND FUTURE WORK

We proposed a scheme for joint compression of multiple overlapping video sequences. We also considered the case when these video cameras are a part of a network and provided algorithms for jointly optimizing both the communication and the compression efficiency at the same time. In the entire design process care was taken to ensure that the joint compression scheme reduced to the MPEG coding scheme for the case of a single video sequence, which we think is an important desirable attribute for any multi video coding algorithm. We are currently working on a real-time ditributed video coding architecture for a network that has 9 pan-tilt-zoom surveillance cameras.

6. REFERENCES

[1] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. 19, pp. 471–480, 1973.

[2] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: Theory, Algorithms and Scaling Laws," *IEEE Transactions on Information Theory*, 2005.

[3] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, pp. 71–83, 2005.

[4] R. Puri and K. Ramchandran, "Prism: a new robust video coding architecture based on distributed compression principles.," *Proc. of Allerton Conference on Communication, Control, and Computing,*, 2002.

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

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

[7] B. Song, A. Roy-Chowdhury, and E. Tuncel, "A multi-terminal model-based video compression algorithm," *ICIP*, 2006.

[8] N. Gehrig and P.L. Dragotti, "Different: Distributed and fully flexible image encoders for camera sensor networks," *ICIP*, 2005.

[9] B. Song, O. Bursalioglu, E. Tuncel, and A. Roy-Chowdhury, "Towards a multi-terminal video compression algorithm using epipolar geometry," *ICASSP*, 2006.