HYPERCUBE BASED INTER VIEW PREDICTION FOR MULTI-VIEW VIDEO CODING

Hari Kalva and Borko Furht
Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL, USA

ABSTRACT

Multi-view video coding (MVC) is being standardized by the MPEG committee. One of the important factors is improving the prediction efficiency by exploiting the spatio-temporal redundancies among the views. The prediction structure has to be selected to balance prediction efficiency as well as coding complexity. Increasing the number of reference views will improve the prediction but also increases the encoding and decoding complexity. A secondary goal is to compactly represent the prediction structure and the dependencies. In this paper we present a hypercube inspired prediction structure for inter-view prediction. The proposed structure balances the efficiency of prediction and coding complexity well. We present the results of motion estimation efficiency and compare it with a 'grid' prediction model. We show that using a Hypercube based model will improve the overall efficiency of the system.

1. INTRODUCTION

Video coding technologies have matured sufficiently over the last few years to make possible new generation of video applications. Multi-view video coding (MVC) has been receiving significant attention among researchers [1-4]. Multi-view video coding (MVC) is also being standardized by the MPEG committee [5-6]. The goal of MVC is to allow coding of multiple camera views such that the user has the freedom of choosing the view point. The biggest challenge here is in developing compression technologies that can exploit the redundancies among the multiple views to achieve high compression ratio. The compression algorithm should also support fast decompression and playback of the selected views at the receiver.

Figure 1 shows a multi-view coding system. Multiple cameras are used at the sender to capture a scene. A multi-view video encoder is used to compress the output of all the cameras jointly. The compressed stream is delivered to a receiver over networks. The receiver can be a TV without multi-view capability, a 3D TV system, or a multi-view receiver with interactive view selection; the receiver is responsible for extracting the views suitable for the receiver. Multi-view coding is an active area of research. An overview of algorithms submitted to the MPEG committee is presented in [6]. The existing technologies that can be applied to multi-view coding are reported in [2,7]. The prediction algorithms that have been evaluated so far create long dependencies and do not scale well with the number of views. The proposed prediction structure, which is based on a hypercube configuration, results in minimized dependencies among the views and at the same time allows exploitation of the redundancies efficiently. The proposed prediction structure can be conveyed to the receiver using a very compact notation to identify the views and dependencies thus resulting in the transmission of fewer bits to the receiver.

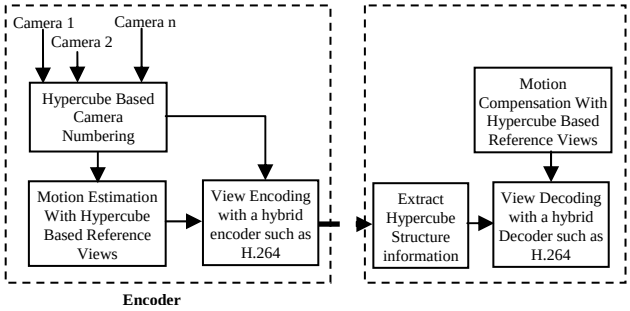

Figure 1. Multiview Coding System

2. HYPERCUBE PREDICTION STRUCTURE

Responding to MPEG's call for evidence, a number of companies have submitted preliminary work on multi-view coding. The proposed algorithms exploit the redundancies among the views by using inter-view reference frames; i.e., frame from a different view are used to predict video in a given frame/view. A number of prediction structures were proposed. Figure 2 shows an example of sequential prediction structure, referred to as 'grid' prediction in this paper. Our experiments compare the 'Grid' and Hypercube prediction for 1D and 2D camera arrays.

The disadvantage of the Grid prediction structure is that it results in sequential dependencies that lead to a complex decoder. In general, if there are n cameras, which means n different views, in order to play the view n, (n-1) views must be decoded. In the prediction structure example shown in Figure 2, 3 previous views must be decoded in order to play the view 4. Table 1 summarizes reference

views that need to be decoded for each view. As the number of cameras increases, these proposed structures create very large dependency chains resulting in huge computational cost for extracting the corresponding views at the receiver. A scalable prediction structure is necessary to minimize these dependencies.

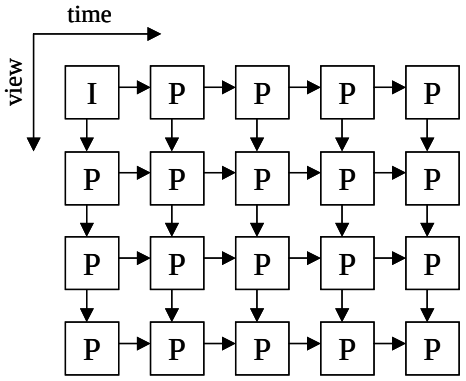

Figure 2. "Grid" Prediction Structure

Table 1: View Dependencies for Grid Prediction

View No. (V_i)	Reference Views (V_j)
0 (000)	-
1 (001)	0
2 (010)	0,1
3 (011)	0,1,2
4 (100)	0,1,2,3
5 (101)	0,1,2,3,4
6 (110)	0,1,2,3,4,5
7 (111)	0,1,2,3,4,5,6

The Hypercube topology was widely studied in the context of parallel processing on multiprocessor architectures because of its simplicity and ease of routing [8]. The same properties of the Hypercube also make it a good candidate to represent prediction structures. A Hypercube is typically described using the number of dimensions. An n-dimensional cube has 2^n nodes. A 3-cube is a regular cube with 6 sides and a 4-cube is usually referred to as a Hypercube. In this paper we use a 3-cube for the sake of simplicity and ease of description. The prediction tools developed can also be applied to higher dimensional cubes.

The main concept is treating each node of the hypercube as a camera view and using the routes between nodes to determine the prediction dependencies. Each camera view can be considered as a node of a hypercube of n dimensions. The numbering of the nodes in a hypercube allows a well structured dependency description. This concept is described below for an 8 camera array, structured as a 3-cube, as illustrated in Figure 3. Figure 3 also shows an example of how the views can be coded; the first frame of view 0 is coded as an I-frame and the remaining frames of view 0 and all other views are coded as P frames. This is similar to the Grid prediction where there is only one I frame. Node 0 is considered the root node and the corresponding view is coded without inter-view prediction. Nodes 1-7 can use inter-view prediction.

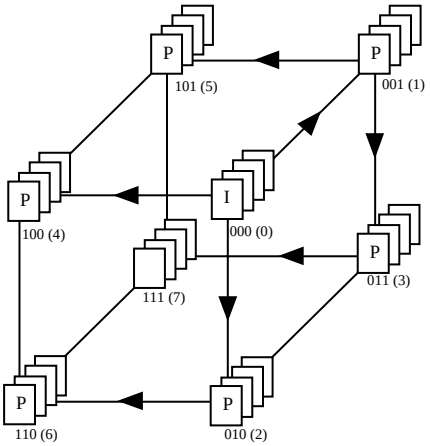

Figure 3. Hypercube with 8-nodes

The nodes in an n-cube are identified using an n-bit binary number. For a 3-cube, the node ID numbers range from 0 to 7. The node with ID=0 is called a root node and this root camera view (root view) is encoded without inter-view prediction. This allows systems without MVC support to just decode and display the root view to the user. The node IDs of an n-cube has the property that the binary IDs of adjacent nodes differ only in one-bit position and each node has n-adjacent nodes. The binary node IDs with the equivalent decimal ID in parenthesis is shown in Figure 3. Any node of a hypercube can be reached from any other node by traversing through the adjacent nodes. We use a distance metric to select a path between a given node and the root node. The sum of the node IDs on a path is used as the distance of the path between the nodes. The reference views for a node V_i can be determined by finding a shortest path from the root (node 0), through the adjacent nodes, to the node V_i; the intermediate nodes on the path form the reference views. Using this approach, the reference views can be readily determined at the receiver without any additional information. For example, node 7 can be reached from node 0 over paths: 0 – 1 – 3 – 7 or 0 – 2 – 3 – 7 or 0 – 2 – 6 – 7 or 0 – 1 – 5 – 7. The shortest path however, is 0 – 1 – 3 – 7 and hence the view 7 can be coded using references frames from additional views 0, 1, 3. Other paths can be used if the selection is explicitly conveyed to the decoder. Table 2 summarizes the view dependencies for a 3-cube structure. In general, in order to decode and play the view n, the maximum number of views to be decoded is $\log_2(n)$. For example, in the 3-cube networks,

where n=8, for the view 7 the number of views to be decoded is $\log_2 8=3$. For a 4-cube with 16 cameras, for the last node 15 to be decoded, there is a maximum of $\log_2 16 = 4$ views to be decoded.

Table 2: View Dependencies in Hypercube Prediction

View No. (V_i)	Reference Views (V_j)
0 (000)	-
1 (001)	0
2 (010)	0
3 (011)	0,1
4 (100)	0
5 (101)	0,1
6 (110)	0,2
7 (111)	0,1,3

2.1 Camera Positioning and Node Assignment

If the number of cameras is greater than 8, a larger hypercube can be created. If the number of nodes is not a power of 2, the higher order nodes will be ignored. For example, if a MVC system has 6 views, this can represented using a hypercube of dimension 3 but without view assignment to nodes 6 and 7.

The cameras in the real world will not be positioned as a hypercube but more as a rectangular array. The cameras have to be assigned node IDs in such a way that the adjacent nodes in fact represent camera views that are closely correlated. The view IDs can be updated as necessary by specifying the updates. Such view ID updates can be indicated in headers of coded bitstreams. An example node numbering for cameras positioned as an array of 8 cameras arranged in a 2D and 1D array are shown in figures 4 and 5, respectively. The hypercube node ID with the view number in the parenthesis is also shown. Such a node assignment is performed at the encoder and preprocessing tools can be used to determine a node assignment that increases the prediction efficiency.

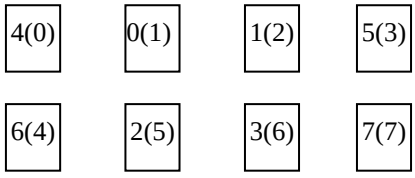

Figure 4. Node IDs of 8 camera 2D array

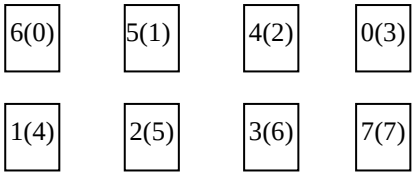

Figure 5. Node IDs of 8 camera 1D array

5. RESULTS AND DISCUSSION

An experimental system was developed to evaluate the performance of the hypercube prediction. The performance was evaluated using two metrics 1) mean of the minimum mean absolute difference (Mean MAD) of blocks in a frame 2) mean motion vector length. Block size of 8x8 is used for motion estimation and full search with a search range of 7 was used. While quantization has impact on motion estimation, it is disregarded in these experiments for the sake of simplicity. The relative performance of the prediction methods evaluated will hold even with quantization. The MPEG MVC test sequences were used for the experiments [5]. The experiments are conducted using the 'Ballroom' sequence captured using a 1D 8-camera array and the 'Akko and Kayo' sequence was captured using a 2D array. Views 26-29 and 46-49 of the 'Akko and Kayo' sequence were used. The view mappings showed in Figures 4 and 5 were used in the experiments. No view pre-processing was done before motion estimation.

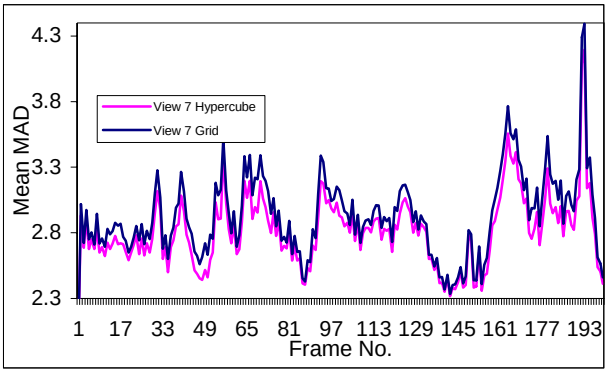

Figure 6. Mean MAD for view 7 of Ballroom with hypercube and Grid predictions

Figure 6 shows the Mean MAD for view 7 of the Ballroom sequence. As shown in Figure 6, hypercube gives a better prediction while requiring only 3 views to be decoded. Figure 8 shows a similar result for the Akko 2D array. For Grid prediction, the 2D array is rearranged as a 1D array in the order 26,46,47,27,28,48,49,29, as shown in the figure 7, to increase the correlation among the dependent views. For hypercube, the node assignment shown in figure 4 was used.

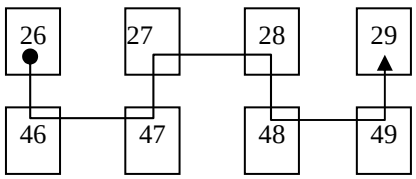

Figure 7. Akko camera 2D array for 'Grid'

Figure 9 shows the prediction for 4 views of Ballroom. View 1 is coded as a root view and has higher MAD on the average as it does not use inter-view prediction. View 7 uses the most reference views (3) and has the least

MAD on average. MAD for view 7 is higher from frame 60 to 120. This could be because of reduced correlation. Further study is needed to understand the anomaly.

The hypercube based prediction offers a flexible structure that can be customized through node assignment and path selection to increase the prediction efficiency. The results reported in this paper are preliminary and have not explored all the possible options. These preliminary results are promising and we expect significant improvements in the performance as the hypercube topology is fully explored and exploited. Pre-processing at the encoder can be performed more effectively using the hypercube based prediction as dependencies are minimized.

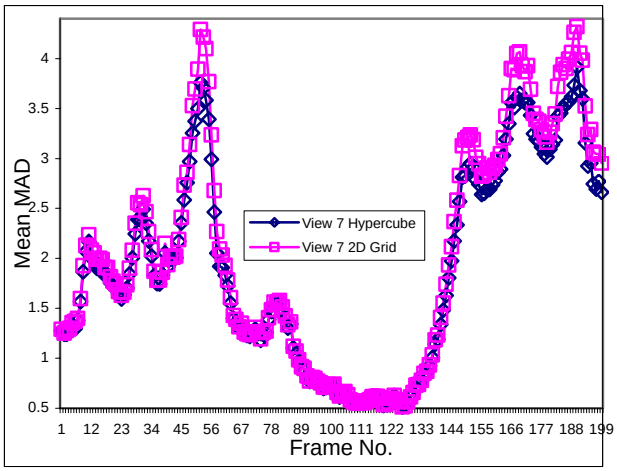

Figure 8. Mean MAD for view 49 (7) of Akko with Hypercube and Grid predictions

5. CONCLUSION

We introduced a hypercube based prediction structure for multi view coding. The proposed prediction structure balances the view dependencies and prediction efficiency well. Furthermore, the prediction structure can be represented compactly for encoding in a bitstream. The structure also scales well with the number of cameras. The efficiency of the structure was experimentally verified and compared with the Grid prediction structure. The results show that the hypercube prediction performs better than the grid prediction and also reduces the view dependencies.

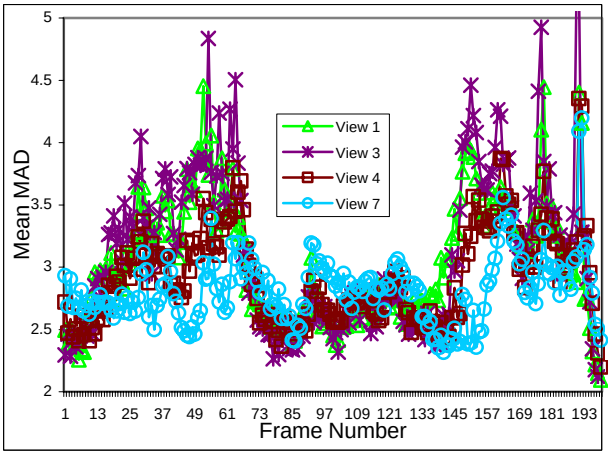

Figure 9. Mean MAD for 4 views of Ballroom with Hypercube prediction

6. REFERENCES

1. O. Schreer, P. Kauff, and T. Sikora, edts., "3D Video Communications, " Wiley 2005.
2. T. Matsuyama, "Exploitation of 3D video technologies," Informatics Research for Development of Knowledge Society Infrastructure, 2004.ICKS 2004.International Conference on, pp. 7-14.
3. M. Ollis and T. Williamson, "The future of 3D video," Computer, vol. 34, pp. 97-99, 2001.
4. A. Smolic and P. Kauff, "Interactive 3-D video representation and coding technologies," Proceedings of the IEEE, vol. 93, pp. 98-110, 2005.
5. ISO/IEC JTC1/SC29/WG11, "Call for Proposals on Multi-view Video Coding (MVC)," MPEG Document MPEG2005/N7327, July 2005.
6. ISO/IEC JTC1/SC29/WG11, "Survey of Algorithms used for Multi-view Video Coding (MVC)," MPEG Document MPEG2005/N6909, January 2005.
7. A. Smolic, K. Mueller, P. Merkle, T. Rein, M. Kautzner, P. Eisert and T. Wiegand, "Representation, coding, and rendering of 3D video objects with MPEG-4 and H.264/AVC," Multimedia Signal Processing, 2004 IEEE 6th Workshop on, pp. 379-382.
8. F. Harary, J. P. Hayes, and H.-J. Wu, "A survey of the theory of hypercube graphs," Computers & Mathematics with Applications, Volume 15, Issue 4, 1988, Pages 277-289.