

# Performance Evaluation of the Hypercube Based Prediction Algorithm for Multi-View Video Coding

Lakis Christodoulou, and Takis Kasparis

Dept. of Electrical Engineering & Information Technology,  
Cyprus University of Technology

Hari Kalva

Dept. of Computer Science & Engineering  
Florida Atlantic University

## ABSTRACT

*Multi-view Video Coding (MVC) is showing a new demand in the video communications, video surveillance systems, video teleconferencing, TV communications, and 3D video games. New algorithms are needed to improve video compression and reduce the complexity of multiple views in order to improve the MVC systems. This paper presents the performance evaluation of the Hypercube Prediction Algorithm (HPA) for MVC. The main objective is to minimize the chain view of the dependencies based on the hypercube structure. Using spatio-temporal predictions based on the hypercube structure we show that MVC compression efficiency can be improved while keeping the dependencies low. The proposed HPA for MVC provides increased flexibility in selecting prediction references by reducing the view dependency by a factor of 2. The performance is compared with a Linear Prediction Algorithm (LPA) that uses one spatial and one temporal reference frame. We show that HPA can be used to allow flexible prediction structures that improve the encoding performance while keeping the sufficient the PSNR video quality.*

**Key Words:** Multi-view, compression, hypercube, linear, dependencies.

## 1. INTRODUCTION

Multi-view Video Coding (MVC) refers to the encoding and decoding of multiple camera views of the same scene [1]. MVC enables new class of video services that allow users to select a view point to watch. MVC has been studied in the early 90s but has seen resurgent interest because of the advances in video compression that makes MVC based systems practical [6]. Other applications of MVC include security, surveillance, and telemedicine. The MPEG committee is currently standardizing MVC. The biggest challenge here is in developing compression technologies

that can exploit the spatiotemporal redundancies in the multiple views to achieve high compression ratios. As the number of cameras used in an MVC application increase, the complexity of the encoding process also increases since the encoder has to consider all the views before making a coding decision. While compression performance can be improved by allowing prediction from all the available views, the decoder complexity is substantially increased as the receiver has to decode all the dependent views of a given view to decode the view.

Our efforts in the area of MVC are focused on developing flexible prediction architectures that can exploit the redundancies across multiple views and across multiple frames. The goal is to develop prediction structures that minimize view dependencies without negatively affecting compression efficiency. Motion vector prediction in MVC is reported in [4] and bit allocation that takes dependent views into consideration is reported in [5]. Our work addresses a more fundamental question: how can we select the right dependent views for predictive coding. We have developed a prediction structure that derives the view dependencies based on a hypercube structure. In this paper we present the HPA and evaluate the performance using the MVC reference software.

Figure 1 shows the general architecture of the multi-view video coding system based on the JSVM reference software used by the MPEG Multi-view standardization group. This architecture supports N views of the same scene and encodes the views by exploiting the large amount of redundancies among the views. We use H.264 as the core compression engine with inter-view prediction to increase compression efficiency.

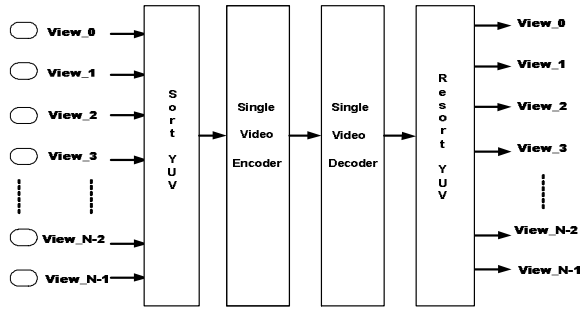


Figure 1. Multi-view Video Coding Architecture

## 2. A NEW HYPERCUBE PREDICTION ALGORITHM

We propose a novel prediction algorithm for MVC that balances the compression efficiency and decoder complexity. The proposed HPA derives prediction dependencies based on a hypercube layout. We have evaluated this prediction algorithm using the JSMV 3.5 software, which is the reference implementation of the MPEG MVC standards effort.

The proposed design in Figure 2 illustrates a new way to implement multiple view cameras encoding and decoding. A prediction algorithm based on the hypercube structure was developed to improve the performance of MVC systems. The camera views are mapped to the nodes of a hypercube and the dependencies are derived based on the position of the camera. The dependencies are communicated to the receiver by providing a node-camera map. Figure 2 shows the nodes of a hypercube and the corresponding camera maps for a 1D array of 8 cameras.

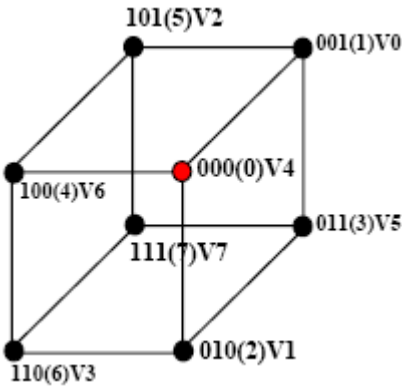


Figure 2. 3<sup>rd</sup> Order Hyper-Cube Algorithm

This type of Hypercube structure allows a well structured dependency description. The binary node IDs, the equivalent n-bit binary number in parenthesis, and the view number  $V_i$  is shown in Figure 1. The reference views for a view  $V_i$  are derived as follows:  $V_i$  can use a view  $V_j$  for prediction if  $V_i$  and  $V_j$  are adjacent and  $V_i > V_j$ .

Table 1 Reference views for an eight camera array

View No. ( $V_i$ )	HPA Ref. Views ( $V_j$ )	HPA No. Views in dependency chain	LPA Ref. Views ( $V_j$ )	LPA No. Views in dependency chain
0 (000)	-	-	-	-
1 (001)	0	1	0	1
2 (010)	0	1	1	2
3 (011)	1,2	3	2	3
4 (100)	0	1	3	4
5 (101)	1,4	3	4	5
6 (110)	2,4	3	5	6
7 (111)	3,5,6	4	6	7
Total		13		21
Average		13/8=1.52		21/8=3.5

Figure 3 shows a basic prediction structure called Linear Prediction Algorithm (LPA) [3]. In LPA, a view forms prediction from a view immediately to its left. Figure 4 shows the prediction dependencies for an 8-camera MVC system using the HPA. The LPA has simple structure where the interview prediction is based on the view to the left. This creates a long dependency chain that increases with the number of cameras. For example, with the LPA prediction, view 7 depends on all the other views. Table 1 shows the reference views, number of views in the dependency chain, and the average number of view dependencies.

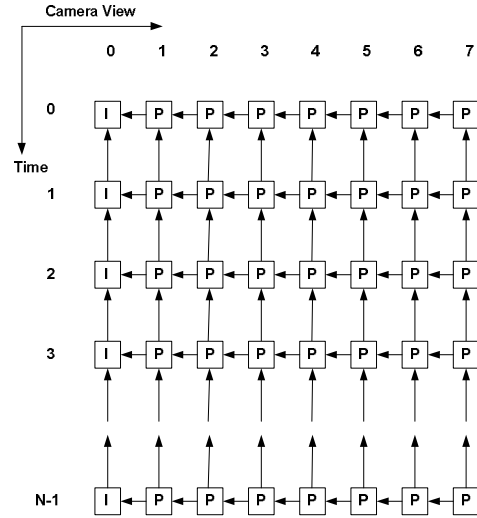


Figure 3. Ballroom Camera View-Time Grid for MVC Linear Prediction Algorithm

The view dependencies for the HPA are derived based on the hypercube node mapping as shown in Table 1. The hypercube structure determines the view dependencies and a camera-to-mapping is done such that the dependent view is strongly correlated. This means view 0 is not mapped to node 0 of the hypercube. Node 0 of the hypercube is a middle-view that can dependent on other views. Optimal view mapping requires taking the camera geometry into consideration. Table 1 also shows the average number of view dependencies (AVD) for LPA and HPA. The AVD metric gives the average number of views that must be decoded to play a view. The smaller AVD leads to lower

complexity decoders but could affect the RD performance. As shown in Table 1, the HPA coded content requires half the number of view on average compared to LPA. The RD performance has to be evaluated experimentally to understand the impact of lower AVD on the RD performance.

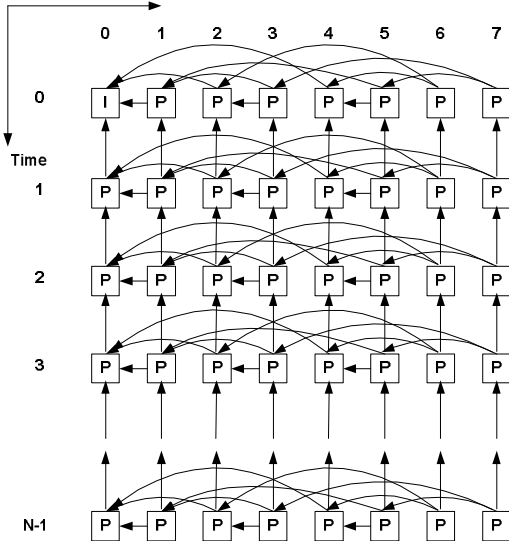


Figure 4. Ballroom Camera View-Time Grid for MVC Hypercube Prediction Algorithm

### 3. EXPERIMENTAL SETUP

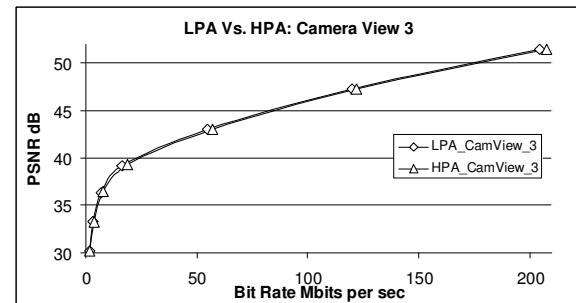
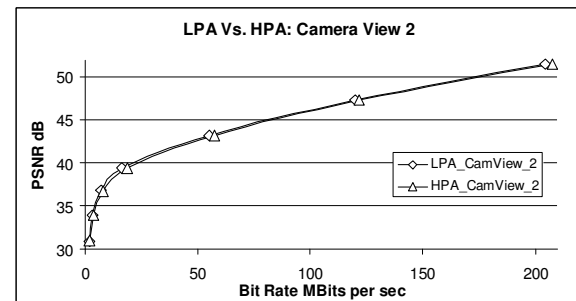
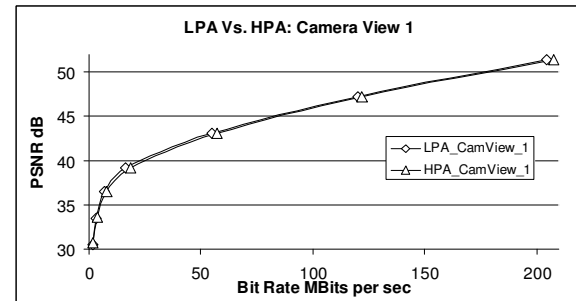
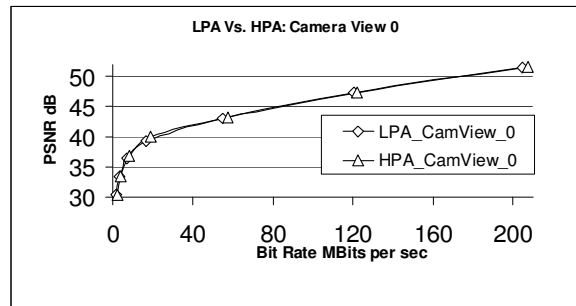
Experiments were conducted to evaluate the RD performance of the HPA and LPA. The experiments were done using JSVM 3.5 modified for multiview coding. The number of reference frames used in the HPA changes with the view and the encoder was modified to change the number of active reference frames depending on the view. The number of reference frames is always 2 for the LPA. The decoded picture buffer size is set to 8 as the prediction uses a frame that is at most 8 frames away. Only I and P frames were used. The reference pictures were set using the reference picture reordering list (RPLR) commands and the format string was written in the JSVM software. The camera to node mapping for this 8 camera arrangement is as shown in Figure 2. Here the mapping of Hypercube nodes to the camera views was chosen such that we obtain maximum prediction improvement and minimize the temporal distance.

For the experiments, we used the Ballroom multi-view video sequences at different QP values. The video sources are 8.33 seconds long, 640x480 resolution, 30 FPS, 250 frames in length, and available in YUV 4:2:0 format. Initially, we started with eight camera views to study the algorithm performance. Only the first frame of view 0 is coded as I frame and the rest are coded as P frames. The first row in Figures 3, and 4 represent the eight camera views at time instant 0, which implies the first picture of the

view to be encoded. The pictures in the same view have temporal dependencies and spatial redundancies are exploited across views.

### 4. RESULTS AND DISCUSSION

We present some experimental results that illustrate the benefits of the HPA algorithm. The LPA with the prediction structure shown in Figure 3 and HPA with prediction structure shown in Figure 4 were both evaluated. Note that because of node-to-camera mapping in HPA, the view 0 shown in Figure 4 actually maps to view 4. The RD plots comparing the HPA and LPA for each view are shown in Figure 5. The bit rate shown is the combined bit rate of all the views.



## 5. CONCLUSIONS

This paper presents a hypercube based prediction structure for multi-view video coding. One of the key challenges in MVC is the complexity of decoding a large number of views. Unless prediction structures are carefully selected, the inter-view prediction could create a long dependency chain that requires large number of view to be decoded to play a given view. The proposed HPA is intended to reduce the prediction dependencies without any drop in RD performance. Both the LPA and the HPA were evaluated using the reference software and the Ballroom video sequence. The results show that the RD performance of HPA and LPA are very similar. The bit rate measured in Mbits per sec for HPA for each camera view remains approximately the same as the LPA. Table 1, shows that the proposed HPA reduces the average number view dependencies by half compared to the LPA leading to simplify more the MVC encoders.

Additional experiments are necessary to study the impact of larger number of views and 2D camera arrangements. The performance of the HPA can be improved by dynamically updating the camera-to-node mapping to maximize the prediction efficiency.

## 6. REFERENCES

- [1] ISO/IEC JTC 1/SC 29/WG11, "Introduction to Multi-view Video Coding", Doc. N9580, January 2008, Antalya, Turkey.
- [2] ISO/IEC JTC 1/SC 29/WG11, "Multi-view Video Coding", Doc. N9978, Hannover, Germany, July 2008.
- [3] Emin Martinian, Alexander Behrens, Jun Xin, Anthony Vetro, Huifang Sun, "Extensions of H.264/AVC For Multiview Video Compression," Mitsubishi Electric Research Laboratories, <http://www.merl.com>, TR2006-048 May 2006.
- [4] X. Guo, W. Gao, and D. Zhao, "Motion vector prediction in multiview video coding," Proceedings of the IEEE International Conference on Image Processing, 2005. ICIP 2005, vol.2., pp. II-337.
- [5] J. H. Kim, J. Garcia, A. Ortega, "Dependent bit allocation in Multi-view video coding," Proceedings of the IEEE International Conference on Image Processing, 2005. ICIP 2005, pp. II- 293-96, 11-14 Sept. 2005.
- [6] H. Kalva and B. Furht, "Hypercube Based Inter-View Prediction for Multiview Video Coding," 2nd International Workshop On Immersive Communication And Broadcast Systems (ICOB), October 2005, Berlin, Germany.
- [7] W. Yang, K. N. Ngan, and J. Cai, "MPEG-4 based stereoscopic and multiview video coding," Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004, pp. 61- 64.

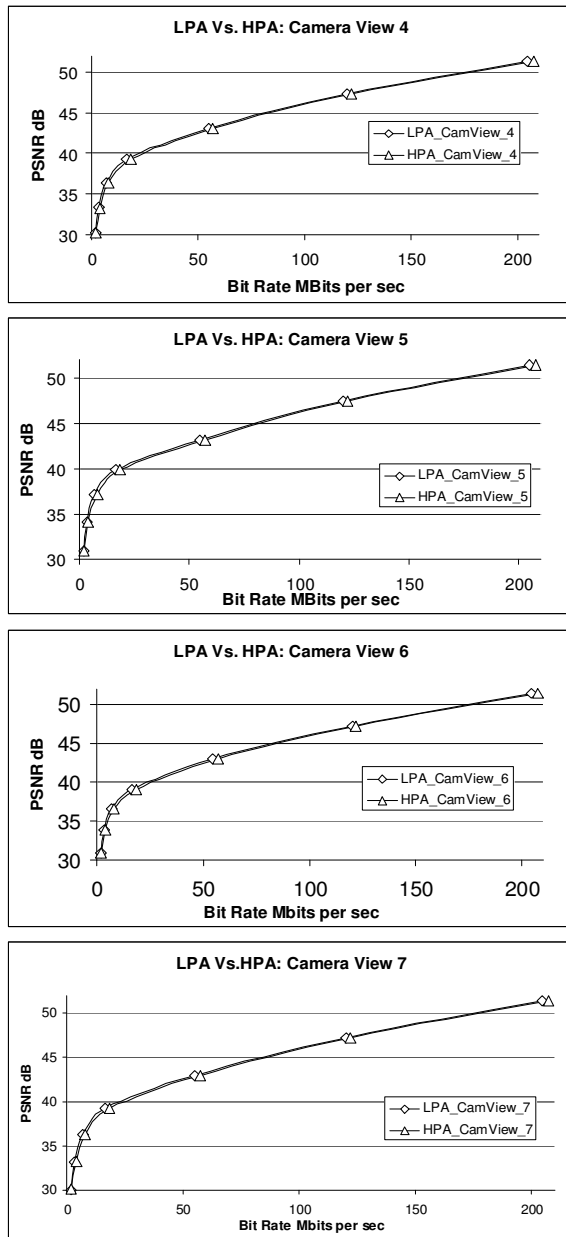


Figure 5. RD performance of the HPA and LPA for the 8 views

Experimental results show that the RD performance of the HPA and LPA is kept at the same PSNR. However, the prediction structure used in the HPA reduces the average number of views to be decoded to half that of LPA. A general observation is that the temporal prediction of the LPA and HPA is kept the same. The main parameter that determines the RD-performance is the spatial prediction. The RD performance of the HPA can be improved by using alternative camera-to-node mappings made possible because of the flexibility of the HPA.