Multiple Camera placement and inter-view prediction in MVC for Virtual Realilty Application

Ranjit Reddy Bayyapu
JNTU College of engineering
Anantapuramu, Andhra Pradesh, India

Ramana Reddy Patil
JNTU College of engineering
Anantapuramu, Andhra Pradesh, India

***Abstract*— Virtual reality (VR) is one of the evolving technologies which is trying to redefine the user experience in entertainment and gaming industries. VR creates a real world for a user to experience and gives the feel of existence, virtually. Many multimedia applications like watching sports, entertainment events and games apart from few interactive applications are considering VR for rendering their content. The videos for the sake of VR content are captured using multiple cameras places at many places in a spherical manner. The 360°scene capture from multiple different directions alone lets the user to experience the scene in all directions horizontally, vertically with top and bottom views. The challenge of coding these multiple views from multiple cameras is being addressed in standards like ITU-T's H.264 extension for Multiview Video Coding (MVC). Though MVC speaks more about the linear arrangement of cameras in a multiview capture of a scene, for VR we can extend the same linear plane of camera placements and subject to the same standard for coding. The MVC standard is expected to evolve for VR kind of applications where cameras also capture the scene with a fisheye lens and not a frame of a video clip. Fisheye lenses gives a wider field-of-view which can be placed across the curve of a sphere. In this paper, the various arrangements of cameras possible on a sphere to capture the scene suitable for making a VR content is explained with possible coding methods where the dependency can be among the closest of the cameras in the same plane of existence.**

Keywords—virtual reality, multiview, camera planes

I. INTRODUCTION

Virtual Reality is a simulated world to give the experience of a real world to a user. It gives the feel for a user as though he is completely immersed in a wonderland with a view of all the directions and all the angles. In applications like VR games, it even makes the user to engage and interact with this virtual world. The applications can grow if the below challenges are addressed:

1. Creation of VR content - Capturing from multiple cameras and the placement of multiple cameras. The placement cannot be in a linear plane to give the experience of watching a scene in 360°. A simpler arrangement can be assuming a cubic placement of multiple cameras, but the true experience from all the angles will be missed out. The cameras that are to be considered also should be different like those with wide field-of-view from fisheye kind of lens.
2. Coding of VR content - Coding of the video frames from multiple cameras, removal of inter-view redundancies and the standard way of building the elementary stream. The uncompressed form of processing the huge size of streams is not a practical solution even it is not a real-time use case. Few of the compression options are discussed in this proposal.
3. Storage of VR content - Standard needs to evolve with the syntax and the tools for coding and decoding. The linear MVC is yet to be complete as a standard and applied in the practical world which enables the options to consider the coding methods for non-linear including the spherical coding of multiview content.
4. Transmission of VR content - As the number of views increase due to the placement on a spherical shape, the content to transmit will be huge.

In [1], H.264's overview is described by T. Wiegand, G. J. Sullivan et.al in which the single view prediction and possible compressions are explained. Multiview standard's extension for high efficiency video coding by K. Muller et al. is given in [2]. In [3], some of the enhancements required to support VR requirements are mentioned though the detailed study of the improvements in various video coding standards is in progress. Higher block sizes like 64x64 and 128x128 are considered and experimental results published which can help in processing of multiple views for VR by reducing the coding time in [4]. The HEVC reference software path is given in [5]. In [6], parallel coding of multiple view is described for MVC to handle 3D video services which can be adopted for VR's multiple views coding. Different planes of the sphere can be subjected to parallel processing. In [7], HEVC's different block partitioning structures are explained by I.K. Kim, J. Min, T. Lee and others. K. K. Sreedhar's Thesis work on Multiview video coding for Virtual Reality in [8] covers a wide range of details from the problems in the existing methods of multiview video processing and the uniqueness of handling multiple cameras to capture a VR content. Kurutepe et.al proposed in [9], a combination of scalable and multiview coding. Base layers are formed from the decoded and reconstructed frames and are used as reference frames which is expected to fail as there would be multiple reference frames needed for decoding and so the complexity in sending these reference frames.

This paper is organized as follows. Section II describes various arrangements of the cameras in order to capture a scene suitable for VR applications. Section III has the possible combinations to select the planes from which multiple cameras would capture the scene suitable to VR and Section IV has the conclusion and future scope.

II. PROPOSED CAMERA PLACEMENTS

A sphere is a three-dimensional geometric object where any point on the circumference is at equidistant from the centre. For a virtual reality kind of application multiple cameras need to be placed around the surface of the sphere to capture the scene in all 360 degrees. The placement of multiple cameras capturing the same scene forms the classic case of multiview video coding. The predominant difference being the ITU-T standards describing the multiview video coding for linear placement of cameras whereas the virtual reality application demands the placement of cameras in a non-linear fashion. Two possible variants of the placement of these cameras on a spherical surface are described below.

a. Horizontal dissection:

A sphere is divided into three planes horizontally. One running across the centre called the top layer, second one at half of the upper hemisphere, called middle layer and the third one at half of the lower hemisphere called bottom layer. Fig.1 shows the division of a sphere in to three layers.

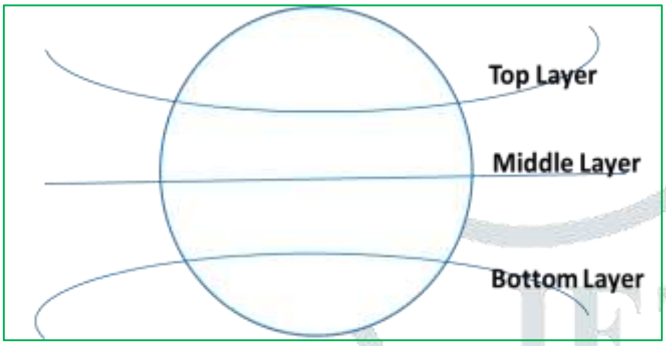


Fig. 1. Horizontal layers on a sphere.

In each layer of this horizontal division four cameras are placed around in a circular fashion adding the number of cameras to twelve. One camera each is placed at the poles and hence the total number of cameras used to capture the scene in 360 degrees becomes fourteen. This arrangement can be varied depending on the purpose of the VR content being captured, like shooting an Advertisement, creating a game content etc. How many ever be the cameras that are placed around the surface of the sphere, the logical separation of a set of cameras into horizontal planes gives the flexibility to view them as though placed on a linear arrangement. This also gives the adaptability with the existing MVC coding tools as per the standard. Fig.2 shows the placement of all fourteen cameras.

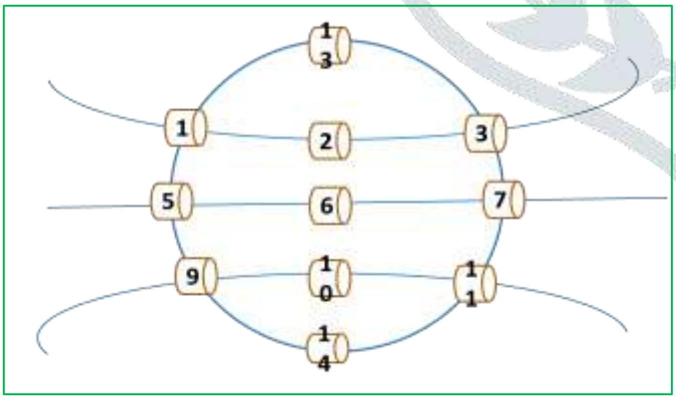


Fig. 2. Placement of multiple cameras in the Horizontal layers.

Such arrangement of cameras helps in visualizing each of the layer as a linear plane which gives the flexibility to code using the standard ITU-T's MVC extension. The three layers form three linear planes as shown in Fig.3.

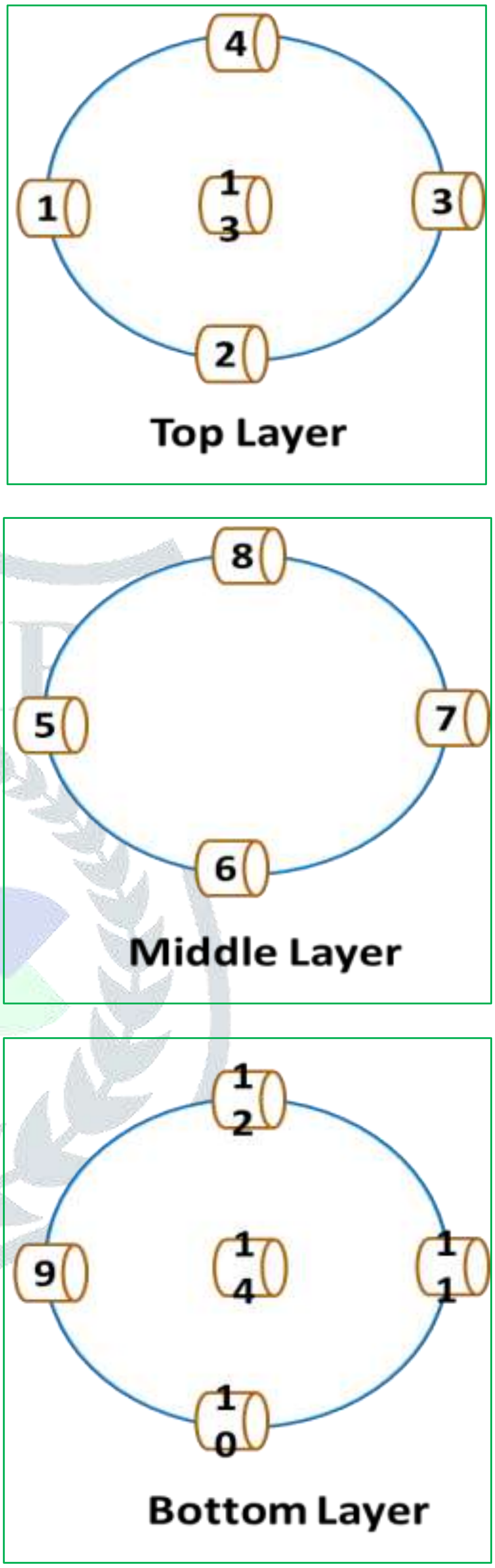


Fig. 3. Placement of cameras in each plane/layer.

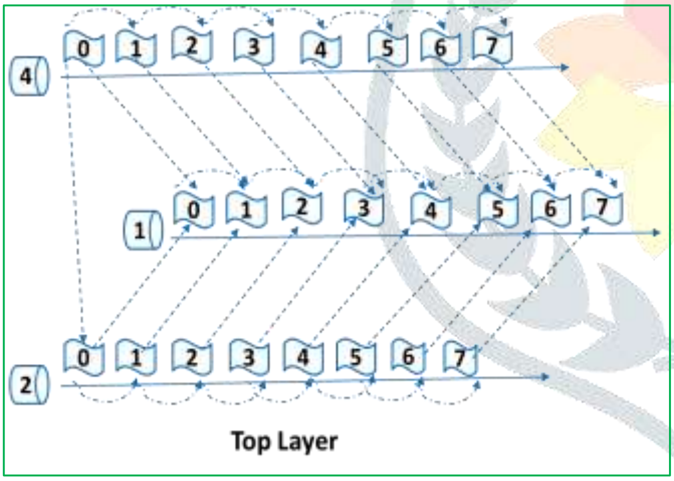
The inter-view prediction can now be applied at each plane level as though the set of four cameras are linearly placed and made to capture the same scene. Fig.4. shows the inter-view prediction structure that would be useful to predict the current frame from its reference views' frame within the plane. As the top layer is a group of views 1, 2 and 4, view1 can be predicted from either view2 or view4 or both apart from the temporal prediction within view1. Similar can be applied from the other side of the sphere where views 2, 3 and 4 can be considered as a pair and view3 predicted from view2 and view4. The top and bottom placed camera positions can be independently coded as they contribute to the important places of the scene that is getting captured. Predicting from cross planes is not discussed in this paper.

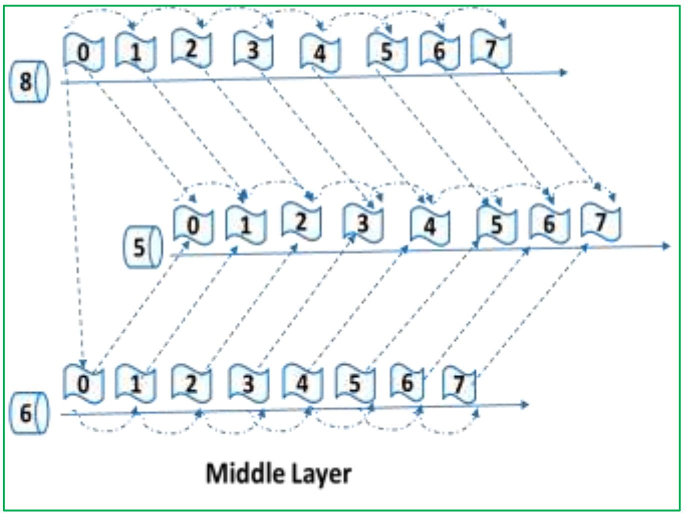
b. Vertical dissection:

A sphere is divided into three planes vertically. One running across the centre called the centre layer, one at the half of the left hemisphere called as the left layer and another at the half of the right hemisphere, called as the right layer. Fig.5 shows the division of a sphere in to three vertical layers.

In each of these layers, a set of four cameras are placed vertically. Four cameras will be those placed on the front part of the sphere, four around the centre and four at the back part of the sphere. Along with these twelve cameras, one camera is placed at the centre of the sphere facing front and one right opposite at the back. Fig.6. shows the placement of cameras on the three vertical layers.

The planes for each set of four cameras will appear similar to the one shown in Fig.3. Inter-view prediction will be limited to each plane and not across planes giving the simplicity to handle. Inter-view prediction can now be applied on each of the plane similar to what is shown in Fig. 4.





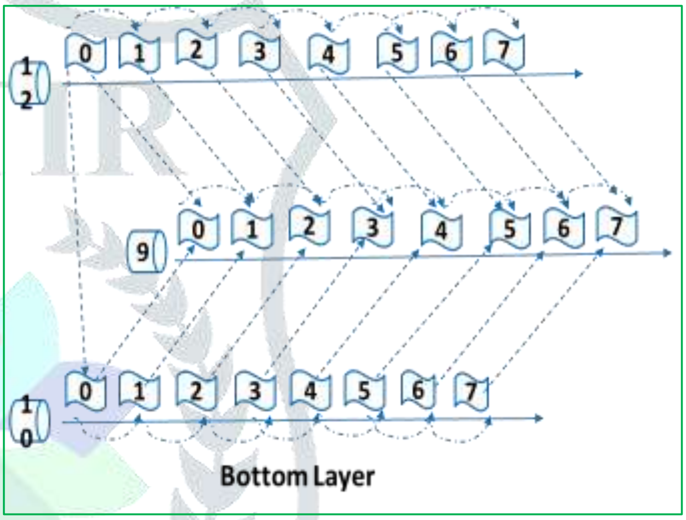


Fig. 4. Inter-view prediction structures in each plane

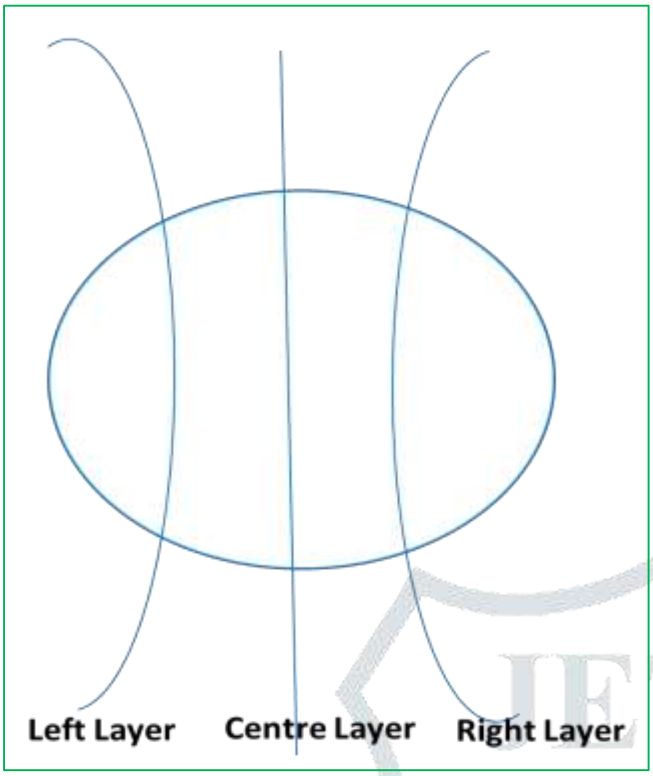


Fig. 5. Vertical layers of a sphere

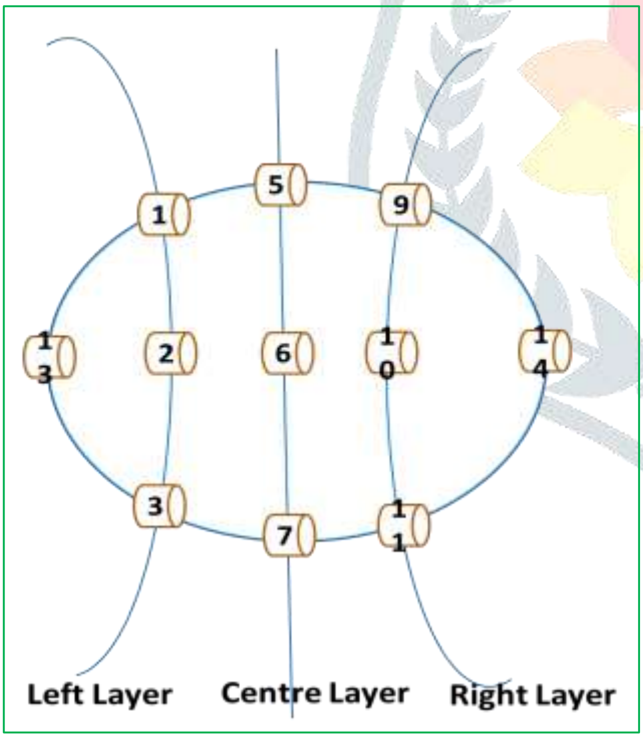


Fig. 6. Placement of cameras in each vertical plane/layer.

If cross plane inter-view prediction is opted for, then some minimal changes will be required in forming the elementary stream as well as at the decoding side to communicate the plane type. These overhead bits will add up to the size of the bit-stream. In the current proposal these are not required to be transmitted as each plane is viewed independently.

III. Conclusion

Virtual Reality is attempting to redefine the user experience of a virtual world where multiple use cases are possible. A user can view the minute details of a newly launched car when a suitable video is placed in a VR device. A gamer can play a soccer game by being in a virtual world of being attacked by players from all the directions. Multiview video coding forms the base of storage and transmission of multiple views to realize these use cases. Creation of content suitable to such games, advertisements is part of contemporary research in respective sectors. This paper presents two of the combinations of placing the set of four cameras across the sphere to capture the same scene which becomes suitable to various VR applications. Apart from the possibility to create multiple views, the inter-view prediction opportunity using the existing MVC standards is also presented. The enhancements required in the standard to address these use cases is also briefed. Usage of fisheye cameras or lenses can give the curved frame of display which will come with challenges in correcting the errors at the edges. Once the standard amendments come in to address such use cases, the possibility to explore cross plane inter-view prediction can also be experimented.

IV. References

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra,“Overview of the H.264/AVC video coding standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no. 7, pp. 560–576, 2003.

[2] G.Tech, Y.Chen, K. Müller, J.R.Ohm, A.Vetro and Y.K.Wang,“Overview of the Multiview and 3D Extensions of High Efficiency Video Coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol.26, no. 1, pp. 35–49, 2016

[3] A. B. Craig and W.R. Sherman, "Understanding Virtual reality", Book, Elsevier, 2003.

[4] M.Narroschke, R.Swoboda, "Extending HEVC by an affine motion model", IEEE, Picture Coding Symposium (PCS), 2013.

[5] H.265/HEVC reference software, Available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/ (March-02-2016).

[6] Y. Chen, Y.-K. Wang, K. Ugur et. al, "The emerging MVC standard for 3D video services,"EURASIP Journal on Advances in Signal Processing, vol. 2009.

[7] I.K. Kim, J. Min, T. Lee, W.J. Han, and J.H. Park, "Block Partitioning Structure in the HEVC Standard", IEEE Transactions Circuits and Systems for Video Technology, VOL. 22, NO. 12, December 2012.

[8] K.K.Sreedhar, “Multiview video coding for virtual reality”, Master of Science Thesis, 2016

[9] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, "Client-Driven Selective Streaming of Multiview Video for Interactive 3DTV,"IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 11, pp. 1558-1565, Nov. 2007