TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

KASHYAP KAMMACHI SREEDHAR

# MULTIVIEW VIDEO CODING FOR VIRTUAL REALITY

Master of Science Thesis

Examiners: Prof. Moncef Gabbouj
Dr. Miska Hannuksela
Dr. Alireza Aminlou
Examiners and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 13th of January 2016

|| ಓಂ ಶ್ರೀ ಗಣೇಶಾಯ ನಮಃ || ಓಂ ಶ್ರೀ ಸರಸ್ವತ್ಯೈ ನಮಃ || ಓಂ ಶ್ರೀ ಗುರುಭ್ಯೋ ನಮಃ ||

# ABSTRACT

**KASHYAP KAMMACHI SREEDHAR**: Multiview Video Coding for Virtual Reality
Tampere University of Technology
Master of Science Thesis, 57 pages
April 2016
Master Degree Programme in Information Technology
Major: Signal Processing
Examiners: Prof. Moncef Gabbouj
Dr. Miska Hannuksela
Dr. Alireza Aminlou
Keywords: Virtual Reality, Video Coding, HEVC, Multiview Coding, Scalable Coding, Fisheye

Virtual reality (VR) is one of the emerging technologies in recent years. It brings a sense of real world experience in simulated environments, hence, it is being used in many applications for example in live sporting events, music recordings and in many other interactive multimedia applications. VR makes use of multimedia content, and videos are a major part of it. VR videos are captured from multiple directions to cover the entire $360^0$ field-of-view. It usually employs, multiple cameras of wide field-of-view such as fisheye lenses and the camera arrangement can also vary from linear to spherical set-ups. Videos in VR system are also subjected to constraints such as, variations in network bandwidth, heterogeneous mobile devices with limited decoding capacity, adaptivity for view switching in the display. The uncompressed videos from multiview cameras are redundant and impractical for storage and transmission. The existing video coding standards compresses the multiview videos efficiently. However, VR systems place certain limitations on the video and camera arrangements, such as, it assumes rectilinear properties for video, translational motion model for prediction and the camera set-up to be linearly arranged.

The aim of the thesis is to propose coding schemes which are compliant to the current video coding standards of H.264/AVC and its successor H.265/HEVC, the current state-of-the-art and multiview/scalable extensions. This thesis presents methods that compress the multiview videos which are captured from eight cameras that are arranged spherically, pointing radially outwards. The cameras produce circular

fisheye videos of $195^0$ degree field-of-view. The final goal is to present methods, which optimize the bitrate in both storage and transmission of videos for the VR system.

The presented methods can be categorized into two groups: optimizing storage bitrate and optimizing streaming bitrate of multiview videos. In the storage bitrate category, six methods were experimented. The presented methods competed against simulcast coding of individual views. The coding schemes were experimented with two data sets of 8 views each. The method of scalable coding with inter-layer prediction in all frames outperformed simulcast coding with approximately 7.9%. In the case of optimizing streaming birates, five methods were experimented. The method of scalable plus multiview skip-coding outperformed the simulcast method of coding by 36% on average.

Future work will focus on pre-processing the fisheye videos to rectilinear videos, in-order to fit them to the current translational model of the video coding standards. Moreover, the methods will be tested in comprehensive applications and system requirements.

# PREFACE

The research work presented in this thesis was carried out from February 2015 - December 2015, at Nokia Technologies Oy, Tampere, in collaboration with Department of Signal Processing, Tampere University of Technology (TUT). During the research work, the author was working as Research Assistant at the of Department of Signal Processing, TUT and as Junior Researcher, in Nokia Technologies Oy.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Moncef Gabbouj, for his support and invaluable guidance constantly throughout this work. I would also like to thank Prof. Moncef Gabbouj for providing me with the opportunity to work in this research project as it has helped me to familiarize with the industrial requirements and academic research.

I am also grateful to Dr.Miska Hannuksela, for his constant technical supervision and immense patience throughout this work, his constructive comments and constant feedback are invaluable and inspiring, which has helped me to perform efficiently.

In addition, I am also very thankful to Dr. Alireza Aminlou, for not only providing me constant feedback and valuable suggestions, but also for motivating and supporting me in my research work. I would also like to thank all my friends in Finland whose company has made these years very memorable one and my friends in India for their moral support.

I would like to thank my spouse, Sushma, for her full support, kindness and patience during these years.

Last but not the least, I am very grateful, from the bottom of my heart, to my parents, Sukanya and Sridhara and my brother Kaushik, for their endless love, devotion and sacrifices for my success and persistent confidence in me.

*Kashyap Kammachi Sreedhar*
*Tampere, April, 2016*

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AMVP | Advanced motion Vector Prediction |
| AVC | Advanced Video Coding |
| BD | Bjøntegaard Delta |
| BL | Base layer |
| BR | Bit Rate |
| CABAC | Context Adaptive Binary Arithmetic Coding |
| CAVLC | Context Adaptive Variable Length Coding |
| CODEC | Coding-decoding |
| CB | Coding Block |
| CTB | Coding Tree Blocks |
| CTU | Coding Tree Unit |
| CU | Coding Unit |
| DCT | Discrete Cosine Transform |
| DVB | Digital Video Broadcasting |
| EL | Enhancement layer |
| FOV | Field Of View |
| GOP | Group of Pictures |
| HD | High Definition |
| HEVC | High Efficiency Video Coding |
| HMD | Head Mounted Display |
| ITU-T | International Telecommunication Union Telecommunication standardization sector |
| JCT-VC | Joint Collaborative Team on Video Coding |
| JVT | Joint Video Team |
| MPEG | Moving Picture Experts Group |
| MSE | Mean Square Error |
| MVC | Multiview Video Coding |
| NAL | Network Abstraction Layer |
| PB | Prediction Block |
| PSNR | Peak Signal to Noise Ratio |
| PU | Prediction Unit |
| QP | Quantization Parameter |
| RD | Rate-Distortion |

| | |
|---|---|
| SAO | Sample Adaptive Offset |
| SNR | Signal to noise Ratio |
| TB | Transform Block |
| TU | Transform unit |
| VCEG | Video Coding Experts Group |
| VLC | Variable Length Coding |
| VR | Virtual Reality |

# 1. INTRODUCTION

Virtual Reality or VR, is a computer simulated medium which gives a sense or experience of complete immersion. It also allows the user to be engaged physically in this simulated environment that is distinct from the real world. These simulated mediums try and imitate the three-dimensional nature of the physical world. This type of flexibility and experience has motivated the use of VR environment in number of areas, for example, gaming, music concert recording and playback, live sporting event streaming, modelling complex and minute mechanical systems, medical diagnosis and in other areas of real world engagement [1].

To bring the sense of complete immersion to the end user, the VR system makes use of visual content, such as stationary images and motion videos. These contents are usually captured from multiple points of view to cover the three-dimensional space of the world. The camera set-up can also vary from linear, cubic to spherical arrangements and commonly cameras with wide field-of-view or fisheye lens are used for content generation. The uncompressed raw videos/images from multiple view points are of high bitrate and thus, makes it impractical for storage and transmission. Videos in VR system are also subjected to certain constraints, such as, varying network bandwidth, heterogeneous mobile devices with limited decoding capabilities and adaptivity based on the current viewing position of the user.

Video compression standards have been used in various application areas including the VR systems, digital video broadcasting (DVB), Blu-ray discs, Adobe flash player, cable and satellite, streaming from internet sources and other areas of multimedia. However, there are certain limitations with the current video compression standards. It assumes videos of rectilinear properties and translational motion model, the camera arrangement to be linear in nature. Improvements to the current standards have been proposed [2]. Although, the proposed methods are efficient, there has been no comprehensive study of the existing video compression standards for the emerging VR system.

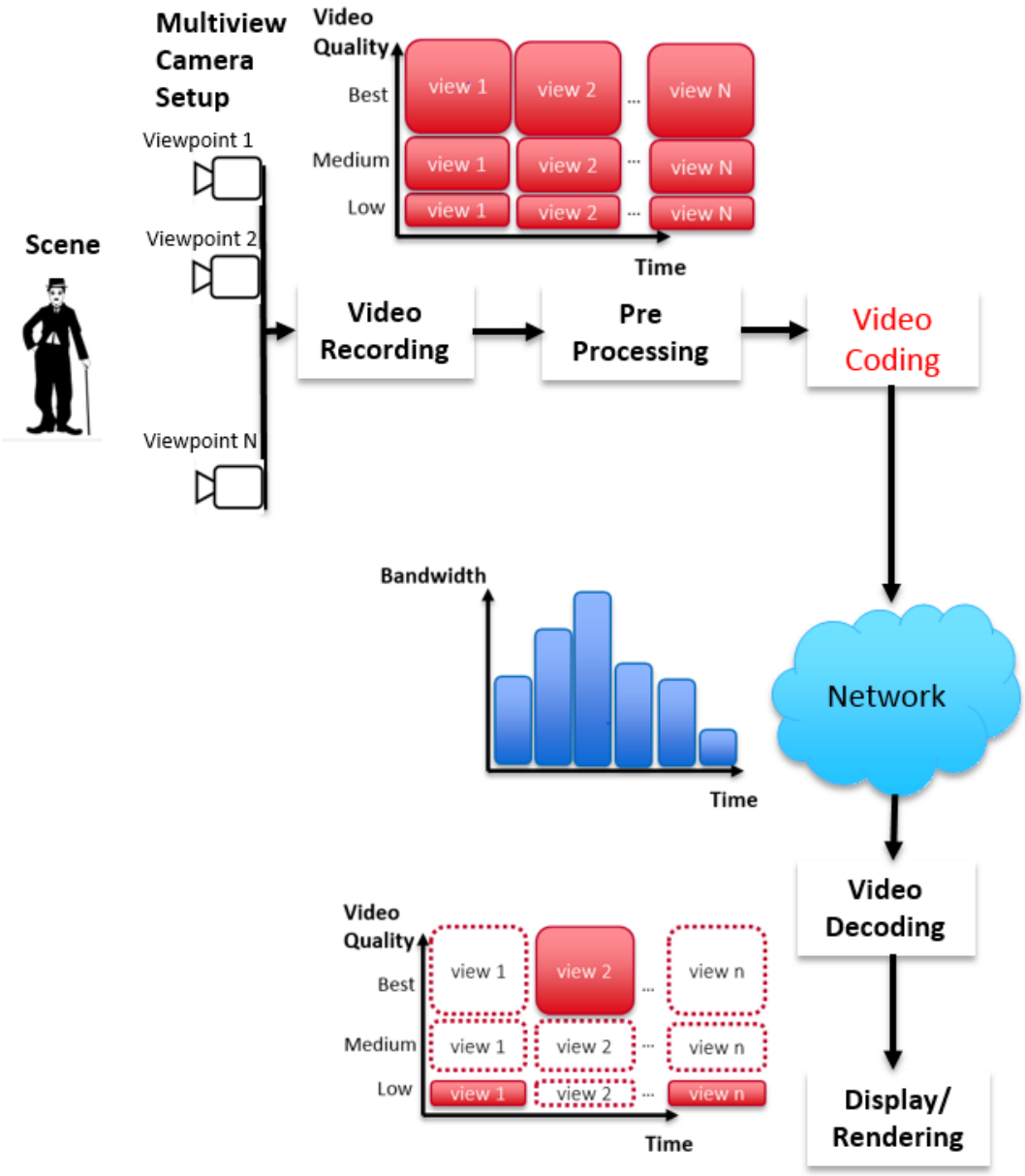


***Figure 1.1*** *A simplified block diagram of the VR system [3].*

The thesis aims to study the existing video compression standards for a spherical multiview camera arrangement producing circular fisheye videos and propose methods that could be used for efficient transmission and storage of multiview videos in a VR system.

A simplified block diagram of the VR system is presented in Figure 1.1. The building

blocks of the system are summarized below.

- **Video Acquisition** - the visual content required for a VR application is usually captured with multiple cameras. The video data is output in a digital form. This process may be temporally and locally decoupled with the steps following it.

- **Pre-Processing** - processing of uncompressed video data with operations such as cropping, reduction of resolution, colour correction, format conversion and/or de-noising, but not necessarily in the same order. More often the same contents with different quality levels are produced in order to support mobile devices with heterogeneous capabilities.

- **Video Coding** - this is the operation of transforming a uncompressed video sequence into a compactly coded bitstream, suitable for storage and transmission for a given application. Several video coding tools and standards are available for compressing the video sequences, from the most widely used H.264/AVC (Advanced Video Coding) [4] [5], its successor H.265/HEVC (High efficiency video Coding) [6] [7] and their extensions to multiview [8] [9] [10] and scalable video coding [11] [12], along with other video codecs such as VP9 [13] [14], Thor [15] and Daala [16].

- **Network** - the infrastructure over which the encoded bitstream is transmitted to the end user. Most often, this encompasses the entire world wide network, the internet. The data transmitted over the network is also prone to errors and variations in the bandwidth availability.

- **Video Decoding** - a hardware/software entity which estimates the original video sequence from the encoded bitstream. In VR applications, video sequences are usually streamed to mobile devices such as smart phones, tablet computers or notebooks with their varying screen sizes and computing capabilities.

- **Display/Rendering** - a representation of video data for viewing at the user end. Even before the display, the decoded video data may be subjected to post processing operations, such as, re-sampling, colour correction and other additional special effects may be added as determined by the application. Certain applications may also employ mechanisms to decrease the overall latency of the application by decoding only certain parts of the video, based on the current view direction of the user.

The processing steps discussed above is only a simplified representation of the overall VR application chain. In many of the real VR applications multiple pre/post-processing steps in collaboration with re-encoding and transmission may be used. These applications may also employ other transcoding techniques, where the incoming video stream is encoded with various video coding standards and with different properties. Furthermore, the VR system has also proposed certain requirements on the video sequences for complete immersion, such as: the video frame rate to be the same as the display refresh rate (eg. >= 75 fps), to reduce perceptible flicker, high resolution videos from full HD, 2K, 4K and beyond, higher field-of-views in the display to match the human visual system [17].

## 1.1 Objective and Scope of the Thesis

This thesis is a part of research study at Nokia Technologies, which aims to investigate the next generation video coding tools for VR applications. The study aims to find the best video coding methods which optimizes the overall bitrate of the multiview videos in both streaming and storage.

The scope of this thesis has been confined by the following factors. The VR applications use multiview videos. The cameras used for video acquisition in the research work produces circular fisheye images. The captured content is overlapping with the adjacent views and hence, a lot of redundant information is present. VR applications also place certain requirements on the resolution and frame rates of the VR video. As indicated in the above section, several video coding standards exist and newer methods/tools can be developed and proposed, however, the aim was to find the methods in existing video coding standards. This led to the choice of international video coding standard tools of H.264/AVC, its successor H.265/HEVC and their multiview and scalable extensions. These standards have been recognized and deployed widely in the multimedia industry. They also provide tools which efficiently transmits the video data in the VR system. The varying network bandwidth, the heterogeneous mobile devices, the viewing directions of the end user are also the most influential factors while designing the video coding methods for VR applications. These elements demand videos of different quality and coding methods of low decoding complexity. At the same time, it is also important to maintain, low system latency and high visual quality at the end user while design these coding methods.

The video data may be subjected to different pre-processing operations before enco-

ding and other post-processing techniques during display. These processing steps and other rendering techniques are beyond the scope of this thesis. In an extension to this thesis work, further investigation for developing newer coding techniques will be investigated.

## 1.2 Thesis Outline

The thesis is organized as below.

**Chapter 2** describes the video acquisition system used for capturing the multiview video content. It is followed with the description of fisheye lens distortion and its impact on the existing video compression standard models.

**Chapter 3** introduces a basic encoder and briefly discusses the existing video coding standards of H.264/AVC, its successor H.265/HEVC and multiview and scalable extensions.

**Chapter 4** discusses the implementation methods used for coding the multiview video. The methods designed mainly aims to reduce the storage and transmission bitrate of the multiview video data. The methods of simulcast, multiview coding, scalable coding and their extension to optimize transmission bitrate have been proposed.

**Chapter 5** is devoted to the presentation of results obtained from the methods discussed in Chapter 4. The simulcast coding technique is used as a reference to compare the rate distortion optimization curve of the designed methods.

**Chapter 6** summarizes the implemented coding methods and suggests direction of future studies.

# 2. VIDEO ACQUISITION SYSTEM

This chapter briefly discusses the video acquisition camera system used for capturing the multiview content. The chapter is divided into two sections, in the first section describes the camera setup. The last section introduces the fisheye distorted images from the camera and discusses its disadvantages in the current video coding standard.

## 2.1 The Camera System

The content used in this thesis is captured with multiple cameras. The camera set up used for content generation is as shown in Figure 2.1.

The camera rig consists of eight cameras on a spherical rig, with each pointing radially outward. All the cameras produce fisheye images/videos covering a field-of-view (FOV) of $195^0$ degrees. Camera 1 together with Camera 4 cover the whole $360^0$ FOV of the scene. While all the other cameras produce overlapping content. Cameras one, two, three and four lie along the equator of the spherical rig, with each $60^0$ degrees apart. Cameras (5,6) and cameras (7,8) are $60^0$ degrees above and below the equator respectively. These camera pairs are aligned along the corresponding longitudes. The camera pairs of (5,7) and (6,8) lie along the same latitude of the spherical rig. Thus, a large portion of the content in camera 1 is overlapping with the adjacent camera pairs of (2,5,7) and similarly the content of camera 4 is overlapping with the adjacent pairs of (3,6,8).

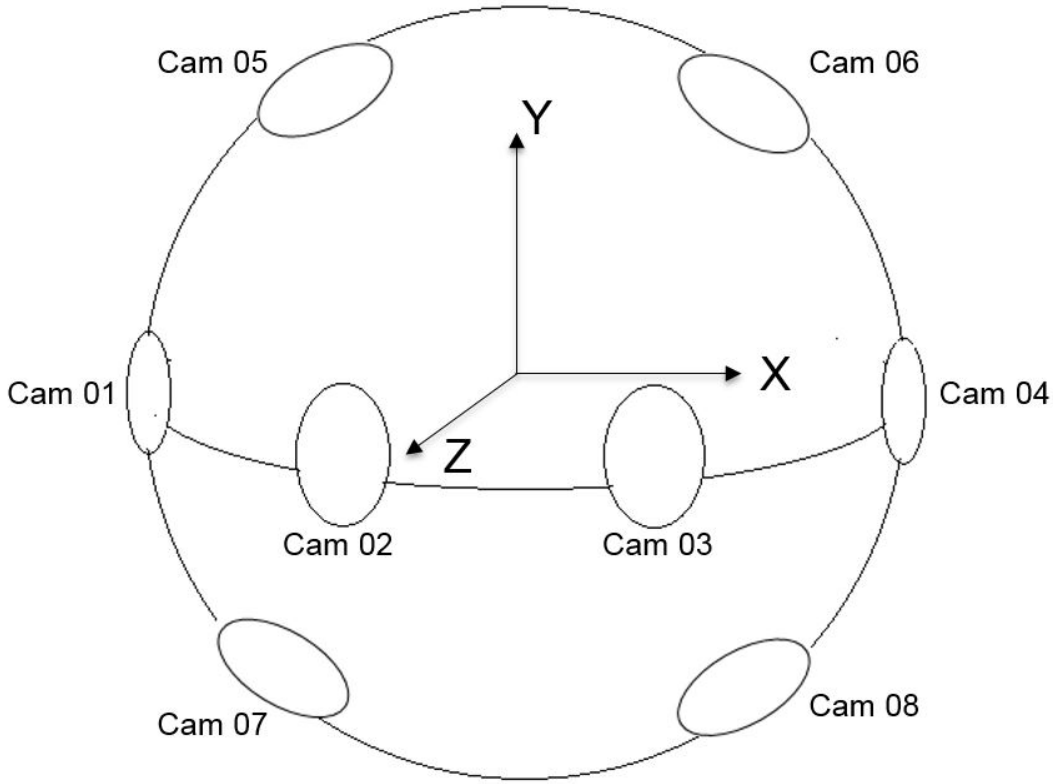


***Figure 2.1*** *The camera system used for content acquisition.*

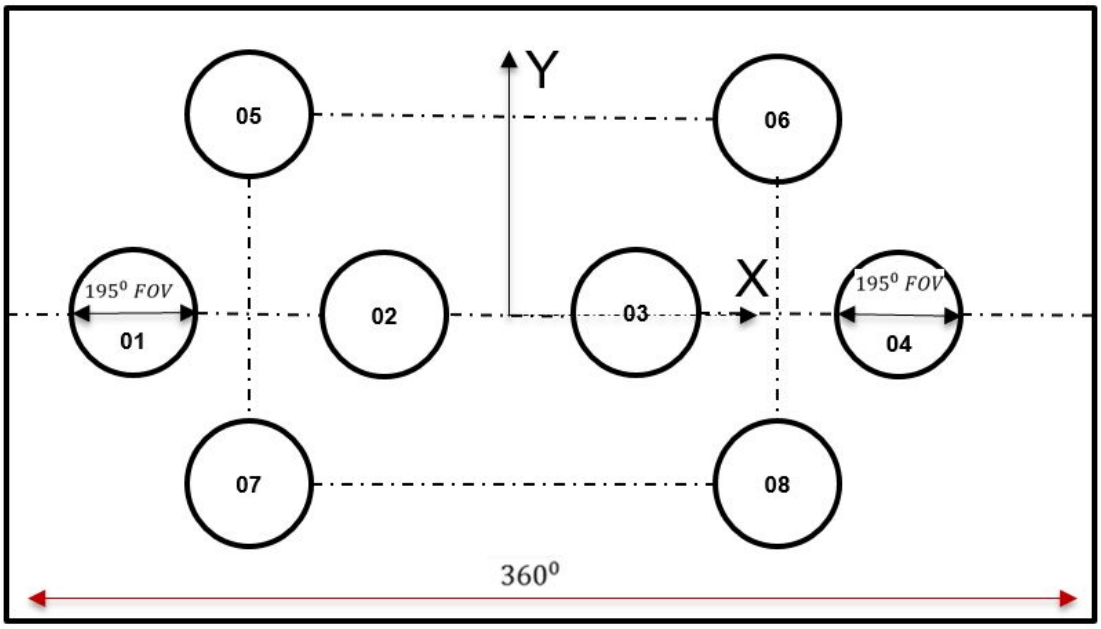


***Figure 2.2*** *The camera system projected on a rectangular grid.*

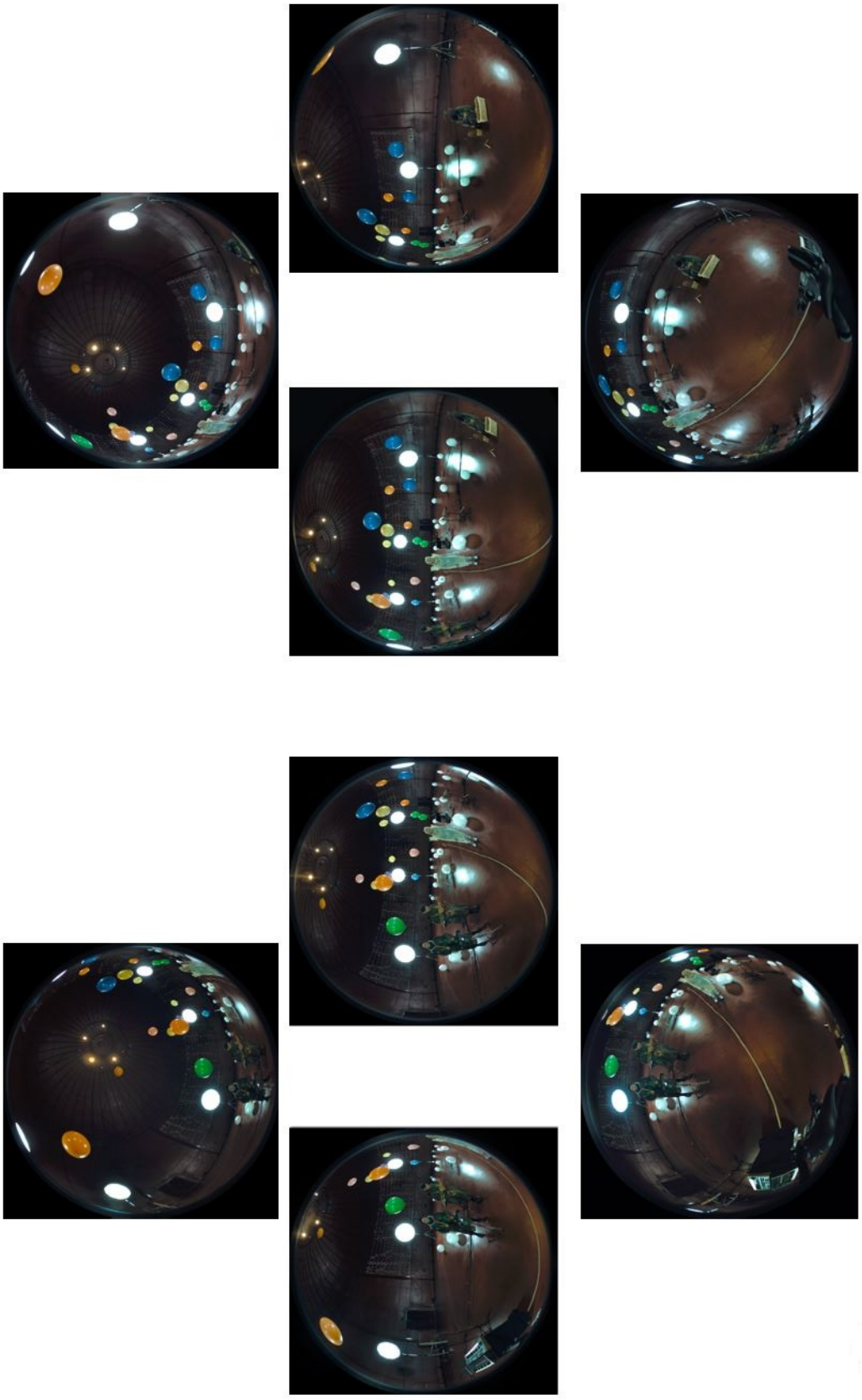

*Figure 2.3 Video frames of the test content from 8 cameras, according to camera layout in Figure 4.8.*

An example video frame from eight cameras used in this thesis is presented in Figure 2.3. The scene was captured at the same time instance in all the 8 cameras. As seen, cameras 1 and 4 capture the scene fully covering $360^0$ degree FOV. The same content information is seen repeated in all other camera frames.

## 2.2 The Fisheye Images

The term '*fisheye*' was christened by Robert W.Wood in his book on Physical Optics [18]. A fisheye camera simulates the fisheye view of the world. They cover a wide field-of-view and can be categorized under wide angle lenses. Such cameras have been increasingly used for capturing panoramic images and videos for VR systems. The images produced by fisheye cameras come in two types: full frame fisheye and circular fisheye. The fisheye lens used in the cameras produce circular fisheye images with $195^0$ degree FOV.

The distortion in fisheye images is usually known as *barrel distortion*. The fisheye cameras completely deviate from the pin-hole camera model. The distortion results in objects at the centre of the image sensor to retain its original shape and as one moves from centre to the sides of the image sensor, the objects are distorted. Figure 2.4, shows the Barrel distortion in fisheye images compared with rectilinear videos shown in Figure 2.5, that is generally output from most of the cameras.

In the standard video coding formats (to be discussed in Chapter 3), a block based motion estimation and compensation is applied. These methods assume a translational motion model, which are suitable for rectilinear videos. The coding methods try to efficiently capture the global motion of objects in the video sequences. However, in case of fisheye distortion, the assumption of translation motion model fails to capture the global and local changes. Many methods have been proposed to reduce the bitrate in these cases, such as, higher order motion model based techniques, which capture non-translational motion, like, the rotation, zoom or deformation of objects in a block [19] [2]. Geometry-adaptive block partitioning techniques, which divides the block using an arbitrary line segment influenced by the motion boundaries in the scene. Practising the above proposed methods would indeed reduce the bitrate of the video sequence, however, in the current thesis, the main aim was to find the methods which are compliant with the existing video coding standards. Thus, in the current thesis, the distortion arising from the fisheye lens is ignored.

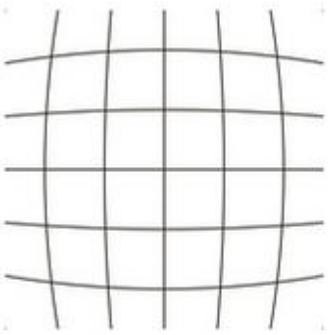

***Figure 2.4*** *Example of Barrel distortion usually found in fisheye images.*

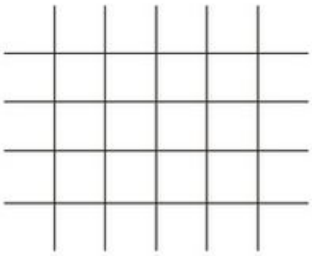

***Figure 2.5*** *Example of the same grid (as in Figure 2.4) in rectilinear images.*

# 3. STANDARD VIDEO CODING TOOLS

This chapter gives a brief overview of the basic video encoder and introduces the international video coding standards of *H.264/AVC*, its successor *H.265/HEVC* and extensions on *multiview* and *scalable* coding. The chapter is divided into five sections, the first section discusses the basic video encoder model which is mostly common to all the above video coding standards with minor changes. The second section introduces the video coding format as defined in H.264/AVC followed by the coding format in H.265/HEVC in the third section. The fourth and final sections describe the principles employed in multiview and scalable extensions of the video coding standards, respectively.

## 3.1 A Basic Video Encoder

Compression is a process of representing data in compact form. Raw uncompressed video data often require large bitrate for storage and transmission. Videos in VR applications are usually captured from multiple viewpoints and have high resolution and frame rates. Thus, it is important to compress video sequences for practical usage.

A video codec models the video into a form which minimizes mainly the temporal, spatial and statistical redundancy. The encoder, compresses the video sequence and the decoder, decodes it to reproduce an estimate of the original video. If the estimated video sequence is identical to the original sequences then, the coding process is lossless; if the decoded video data is different from the original video then the process is lossy [20].

A simplified block diagram of the video encoder is shown in Figure 3.1. The coding process can be categorized into two main paths, the encoding path, where a video frame is encoded and a compressed bitstream is produced. The second is the reconstructed path, where the encoded frame is decoded within the encoder and the

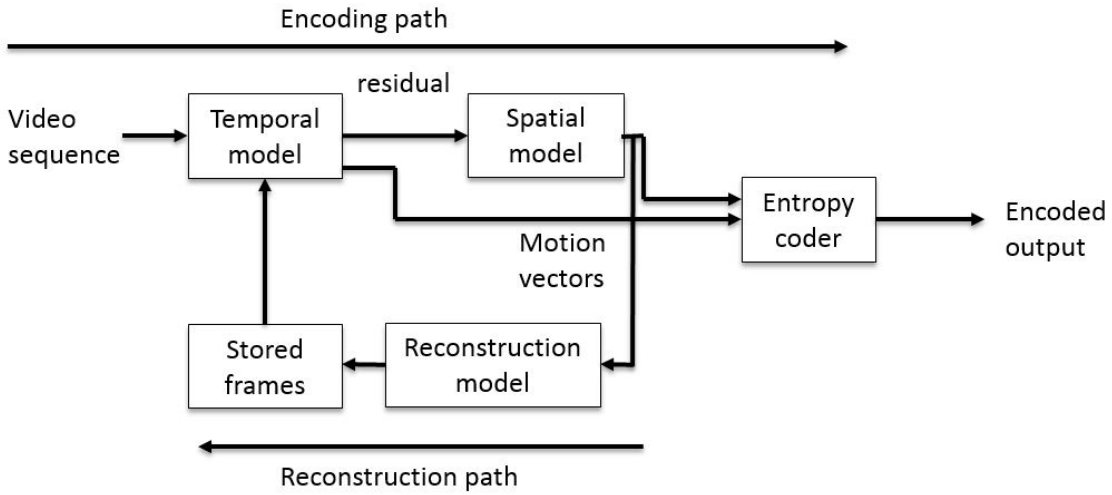


***Figure 3.1*** *A simplified block diagram of a basic video encoder.*

reconstructed frame is used as reference to code future frames.

The video encoding process mainly consists of three functional units: a *temporal model*, a *spatial model* and an *entropy coder*. The following sections describe briefly, the three models of the video encoder.

### 3.1.1 Temporal Model

This model takes an uncompressed raw video frame as input and compresses it by manipulating the temporal redundancy coming from temporally neighbouring frames of the video sequence. This process is also known as *Inter prediction*. The model constructs a prediction of the current frame from the temporally neighbouring frames and outputs a residual frame which is the difference between the prediction and the current frame. Along with the residual it also outputs model parameters called motion vectors. The process of predicting the current frame from temporally neighbouring frames to minimize the energy of the residual is called motion compensated prediction. The motion vectors describe how the motion was compensated.

#### Motion Compensated Prediction

Frames of a video sequence can change in different ways due to camera motion, object motion, lighting changes, scene changes and uncovered regions. These changes

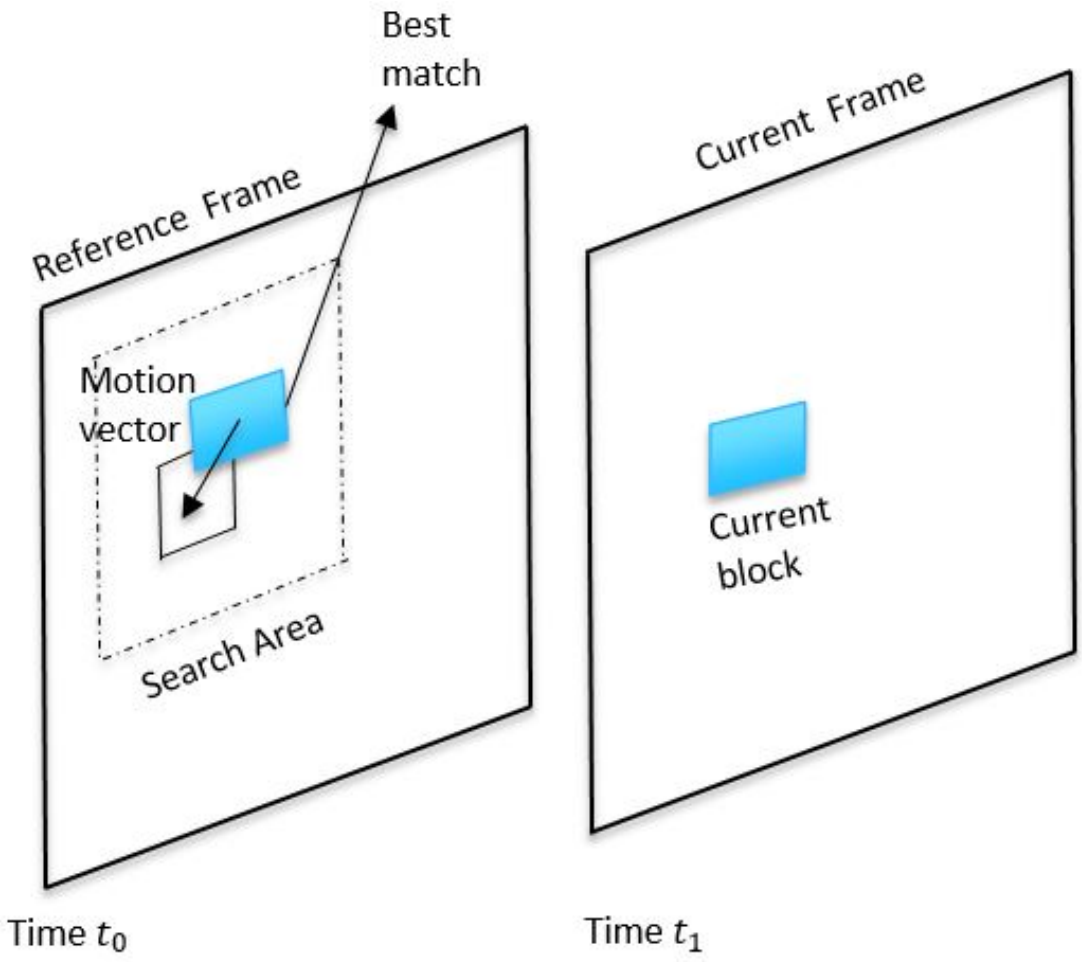


***Figure 3.2*** *Motion estimation.*

can be compensated by predicting the motion between frames. A simple method of temporal prediction, where a temporally neighbouring block of a video sequence is used as a reference frame to predict the current block is shown in Figure 3.2.

In standard video coding tools, the prediction is typically done at block level. This process of predicting and compensating the current frame at block level is called block based motion estimation and compensation. It involves, finding a block of size $N_1 \times N_2$ from the reference frame, which, closely matches with the block of the same size in the current frame. The search for the best match in the reference frame is along a defined area with the centre at the current block position. This search for the best matching block is called motion estimation. The estimation process is further improved by using half and/or quarter pixel resolution during the matching step.

The best matching block of the reference frame is then selected and subtracted from the current block to form a residual block which is encoded along with motion vectors for transmission. The process of producing residuals along with the motion vectors from the best matching block is called motion compensation. The encoded residual is decoded within the encoder and is added to the matching block to form the reference frame which could be used for motion compensation prediction of future frames in the video sequence. This reconstruction in the encoder is necessary to ensure that same reference frame is used in both the decoder and the encoder.

### 3.1.2 Spatial Model

At this stage, the residual data is further de-correlated and is converted to a pattern which can be efficiently encoded with an entropy coder. This stage mainly consists of three components: transformation, quantization and reordering.

- **Intra Prediction** - in this mode a residual is formed by subtracting the current block from a prediction block which is formed based on previously encoded and reconstructed blocks. This mode operates within the current frame being encoded. It exploits the spatial correlation among the pixels.

- **Transform** - in this stage the residual data is converted into the transform domain. The transform function is chosen based on following criteria:

  1. The transformed data must be de-correlated and compact.
  2. The transform must have an inverse function.
  3. It should be computationally tractable.

  Usually, the transform is performed on blocks of size $N \times N$. DCT (discrete cosine transform) and its variants are commonly used in video CODECs.

- **Quantization** - it is a process of converting input data, in a range of values say $R_I$ to a smaller range of output values $R_O$, such that, the output quantized values can be represented with fewer bits than the original. The output of a quantization process is a sparse array of quantized coefficients. The amount of quantization is determined by a critical parameter called the step size $QP$. It is a lossy process (non-invertible), hence, the original data cannot be recovered at the decoder.

- **De-blocking Filter** - the block based encoding structure of the video encoder results in visual artefacts in the form of blockiness. This artefact is removed by a co-called de-blocking filter. It reduces the artefact near block boundaries and prevents propagation of noise.

- **Reordering** - it is a process of efficient grouping of quantized coefficients, which are output from the quantization process. As the sparse array coming out of the quantization step typically has large number of zero coefficients, the reordering helps in representing the zero values effectively. In the video coding standards mentioned above the reordering is often done through the co-called zigzag scan.

### 3.1.3 Entropy Coder

In this process, the elements of the video sequence are represented in a compressed form, which can be efficiently transmitted or stored. The elements may include reordered quantized coefficients, motion vectors coming from motion compensated prediction, headers, markers and other supplementary information. The standard tools used different combinations of variable length coding, VLC, which maps an input symbol to a variable length codeword. In this coding method frequently occurring symbols are mapped to short VLCs, whilst infrequent symbols to long VLCs. The now common and upcoming video standards use more efficient VLC coders such as Context-adaptive variable-length coding (CAVLC), Exponential-Golomb coding and Context-adaptive binary arithmetic coding (CABAC).

### 3.1.4 Group Of Pictures (GOP)

The group of pictures or GOP is a collection of successive pictures in a coded video stream. A full video sequence is usually represented as collection of periodically repeating GOP structure. A simple GOP structure is shown in Figure 3.3 [21].

- **I-picture** - intra coded picture is a frame in the GOP which is encoded independently of other frames. In the decoding order, each GOP usually starts with an intra picture.

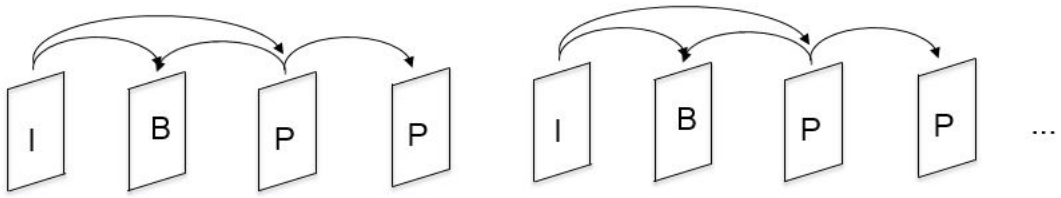


***Figure 3.3*** *Example of a simple GOP structure with two periodic GOPs.*

- **P-picture** - predicted pictures are the ones which are coded with prediction from single reference, also called *uni-prediction*. However, there is no restriction on the reference picture to be in past of the current picture.

- **B-picture** - bi-predictive coded pictures are encoded with prediction by a combination of two references. The reference pictures are typically from both the past and the future of the current picture. The usage of B-picture increases the latency due to complexity of prediction from multiple reference pictures [22].

## 3.2 H.264/AVC - Advanced Video Coding

The H.264/MPEG-4-Part 10 advanced video coding (AVC) was introduced in 2003. It has became very widely used in multimedia industry. The standard was developed by Joint video team (JVT) of VCEG (Video Coding Experts Group) from ITU-T (International Telecommunication Union Telecommunication standardization sector), and MPEG (Moving Picture Experts Group) of ISO/IEC [23].

The standard uses the same basic principles for encoding the video sequence as stated in Section 3.1. The various coding tools which are part of the H.264 encoder are detailed below [24].

1. **Intra-Prediction** - this prediction manipulates the spatial redundancy between the pixels. The processing is done at a macroblock level, which is of size 16x16 samples. The macroblock can be further divided into blocks of different sizes like 8x8, and 4x4, with 4x4 being the smallest block size. This division of blocks is based on, if, the processing is done in luma or chroma samples respectively.

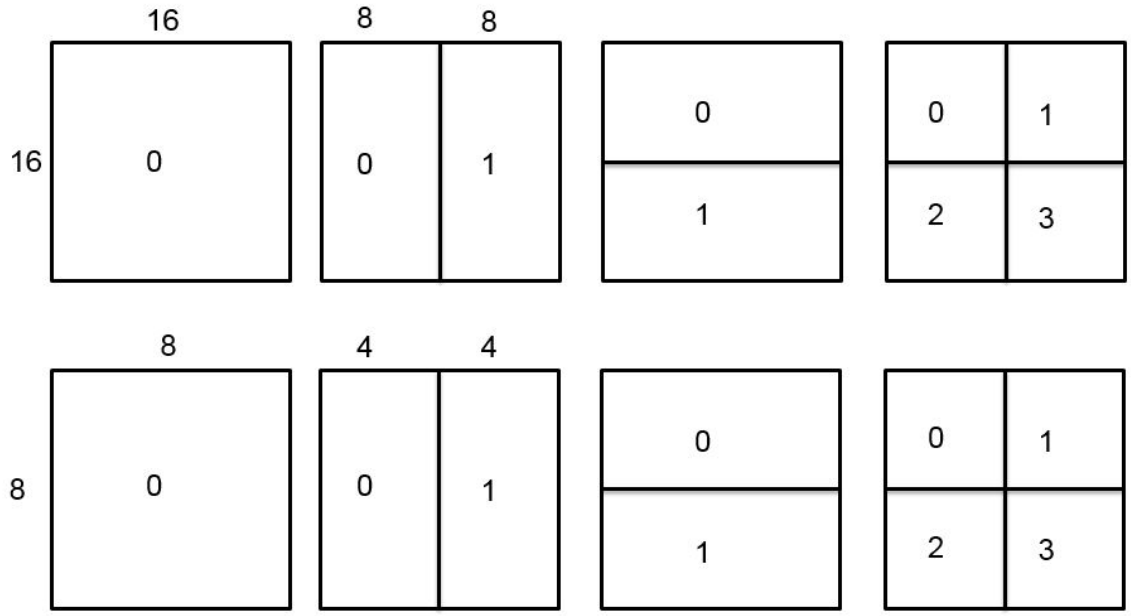


***Figure 3.4*** *Macroblock partitioning in H.264/AVC inter-prediction. At top (L-R) 16x16, 8x16, 16x8, 8x8 blocks. In bottom (L-R) 8x8, 4x8, 8x4, 4x4 blocks.*

2. **Inter-Prediction** - this prediction exploits the temporal redundancy in the video sequence. It uses the block based motion estimation and motion compensation algorithm for the prediction process. The picture is divided into macroblocks of size 16x16. The macroblocks can be further divided into smaller blocks of size 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4. The partitioning of macroblock is shown in Figure 3.4. The smaller block size ensures less residual data; however, it also implies the increase in the number of motion vectors and hence the increase in the number of bits for encoding those vectors. In order to improve the prediction process, it uses sub-pixel motion vectors from half-pixel to quarter-pixel sample accuracy.

3. **Transform and Quantization** - blocks of the residual data is transformed and quantized with a 8x8 or 4x4 integer transform. A modified discrete cosine transform (DCT) is used in the encoder. The output transform coefficients are quantized according to the quantization parameter (QP). It is a number by which each coefficient is divided by an integer value. H.264 uses a hierarchical transform structure, it groups the dc coefficients of the neighbouring 4x4 luma transforms to a 4x4 block. These blocks are transformed again with Hadamard transform. It also uses an in-loop de-blocking filter to remove the blocking

artefact caused by block based transformation and quantization.

4. **Entropy Coding** - this is the final step in the encoder. The inputs to this stage include transform coefficients of the residual data, motion vectors and the other encoder information. The standard uses two types of entropy encoder. The first method is a combination of universal variable length coding (UVLC) and context adaptive variable-length coding (CAVLC). The second method is context-based adaptive binary arithmetic coding (CABAC).

## 3.3 H.265/HEVC (High Efficiency Video Coding)

The H.265/MPEG-H Part 2 High Efficiency Video Coding standard, was was developed by joint collaborative team on video coding (JCT-VC), as a collaboration by the video coding experts group from ITU-T Study Group 16 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG). It is the successor of H.264/AVC and claims to bring a bitrate reduction of up to 50% in comparison [25].

In principle the H.265/HEVC encoder codes a video sequence in a similar manner as in H.264/AVC. However, the bitrate improvement is achieved particularly with the following methods [26] [24].

1. **Coding Tree Unit (CTU)** - The CTU replaces the macroblock structure of H.264. The sizes of CTUs vary from 8x8 to 64x64. These units are partitioned in a quad tree structure. An example of CTU partitioning and the corresponding quad tree structure is shown in Figure 3.5. This type of structuring allows for flexibility in partitioning, while, maintaining design consistency. Every leaf node of the CTU is called a *coding unit* (CU). These units define the prediction type between spatial and temporal schemes. The CU may have several *prediction units* (PU) and *transform units* (TU). The TUs are represented by a quad tree called the *transform tree*.

   CTUs consists of coding tree blocks (CTB) and its associated syntax elements. These blocks specify the two-dimensional sample array of a color component. Thus a single CTU contains one CTB for luma and two CTBs for chroma components. The same arguments are valid for CU, PU and TU, which contain coding block (CB), prediction block (PB) and transform block (TB) respectively [27].

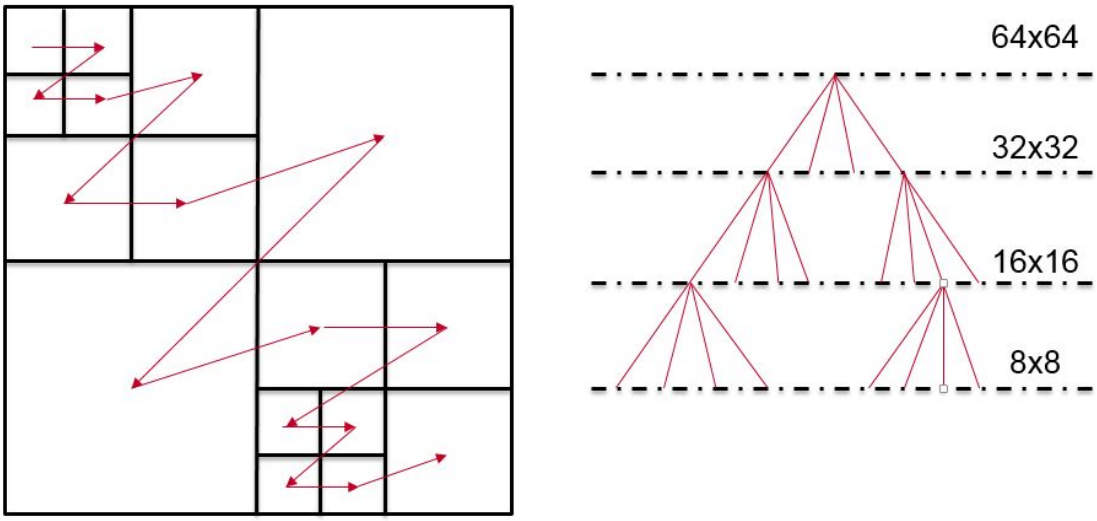


***Figure 3.5*** *To left, Example of CTU partitioning and processing order in HEVC. To right, Corresponding coding tree structure. The minimum CU size is equal to 8x8.*

2. **Intra Prediction** - this mode supports 33 directional modes along with planar and DC prediction modes. The planar prediction helps in generating smooth sample surfaces. Other elements of HEVC intra coding design include: adaptive smoothing of the reference sample, filtering of prediction block boundary samples, mode-dependent prediction residual transform and coefficient scanning and finally coding based on contextual information [28].

3. **Inter Prediction** - the improvements of HEVC over AVC in this prediction mode is as follows. HEVC uses the so-called *merge-mode*, where, motion parameters are not encoded, instead, a candidate list of motion parameters is created from the corresponding PU. Generally, motion parameters of spatially neighbouring blocks and also temporally predicted motion parameters that are obtained based on the motion data of a co-located block in a reference picture. These chosen motion parameters is signaled through an index into the candidate list. Advanced motion vector prediction (AMVP) algorithm is used for prediction. In AMVP algorithm, a candidate list is created for each motion vector. The candidate list may consist of motion vectors of neighbouring blocks with the same reference index and also temporally predicted motion vectors. These motion vectors are coded by signalling an index to the candidate list for specifying the chosen predictor and coding a difference vector. These tools help in coding of motion parameters efficiently in comparison to

previous standards.

4. **Transform and Quantization** - it uses a similar type of transform and quantization schemes as in H.264/AVC.

5. **De-blocking Filter** - The in-loop de-block filtering process has been improved by simplifying the design. This simplification helps in its decision-making and filtering process, thus, making it friendly for parallel processing. Sample adaptive offset (SAO) is added within the inter-picture prediction loop after the de-blocking filter. It is a non-linear amplitude mapping scheme, which helps to reconstruct the original signal amplitudes by using a look-up table.

6. **Entropy Coding** - it uses an improved CABAC coding scheme (similar to the coding method used in H.264). The evolved coding scheme improves the throughput speed mainly for parallel processing, the compression performance and reduces the context memory requirements.

The extension of the standards for multiview and scalable coding exists for both H.264/AVC and H.265/HEVC. However, consider the fact that the video coding standard of H.265/HEVC is the current state-of-the-art in video coding, only the extensions to H.265/HEVC will be considered in the following sections.

## 3.4 H.265/MV-HEVC (Multiview Extension)

In order to address the needs of a broad range of applications which utilize multiview videos, the standardization committee of HEVC proposed for extension of the standard into multiview video coding. This extension enables the representation of multiview and stereoscopic video sequences in a compressed form.

The multiview video coding extension of HEVC (MV-HEVC), is backward compatible for mono-scope decoding. It exploits inter-view redundancy in the prediction process. One of the key aspects of this extension is that the primary block based coding and the decoding process of HEVC remains unchanged. Its fundamental principle is to re-use the underlying 2D coding tools of HEVC, with the changes done only to high-level syntax in the slice header level and above. The goal of high-level syntax principle is met by allowing the inclusion of pictures which originate from direct reference layers in the reference picture list(s), used for decoding pictures of predicted layers, in all other cases these inter-layer reference pictures are

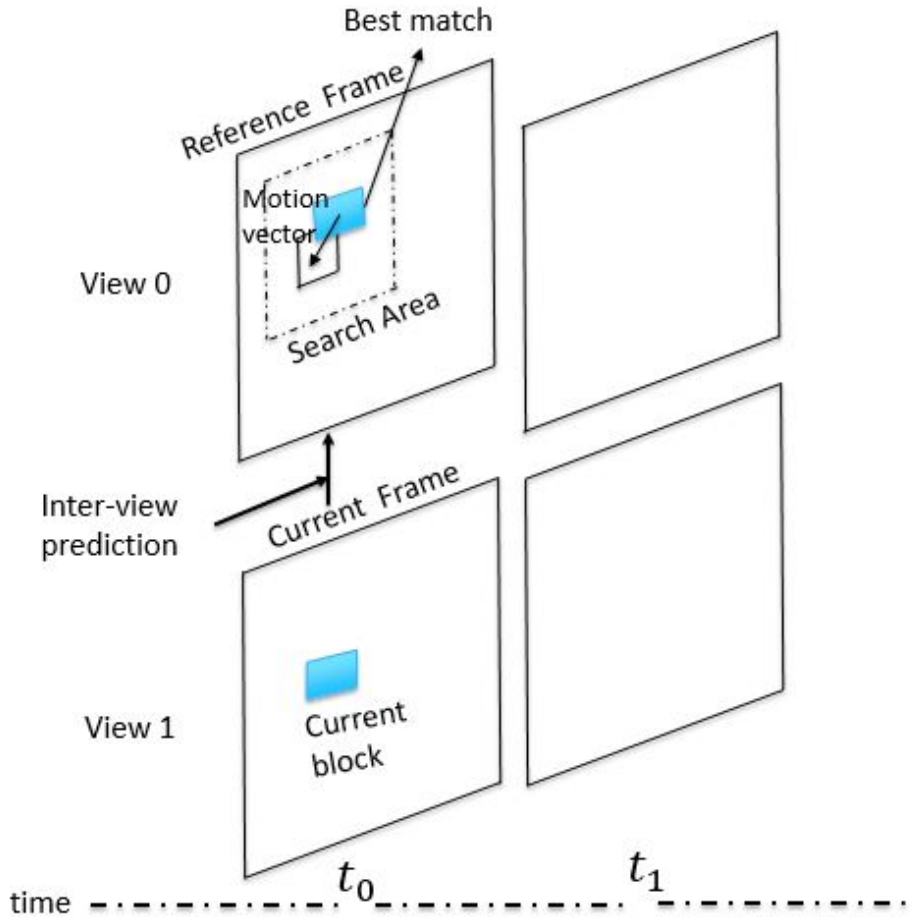


***Figure 3.6*** *Example of motion estimation with inter-view prediction.*

treated identically to any other reference pictures. This design allows, multiple views or the so-called multi-layers to be encoded as different HEVC-coded representations of the video sequence and multiplexed into a single bitstream. The base view being compatible with the standard single layer coding of HEVC in order to enable the extraction of primary views. While the dependencies created by inter-layer prediction to achieve increased compression performance.

Some of the additional high-level syntax include in the MV-HEVC is listed here [29] [30] [9] [10].

1. **Inter-View Prediction** - this method takes advantage of both inter-view and temporal redundancy for compression. A basic prediction structure is shown

in Figure 3.6. With this structure, the reference list is now updated with both temporal and inter-view reference pictures. Among these references, the best predictor, based on rate-distortion cost, will be chosen. Such a design structure helps to retain the block-level coding from HEVC, with changes only to high level syntax elements. The changes in the high level syntax include, among many, the indication of the predictor dependency across views.

## 3.5 H.265/SHVC (Scalable Extension)

Improvements in video compression technology have fueled the use of digital videos in a range of applications and mobile devices. Applications such as video conferencing and video streaming over best effort wired and wireless networks demand for video streams which provide adaptability according to the requirements of the decoders and network conditions. These rising demands have motivated the need for scalable extension of the HEVC standard. Scalability, in this context, refers to a property of video bitstream that allows for removing parts of the bitstream according to the needs of end users and receiving devices. The scalable extension of HEVC allows coding of video sequences in multiple layers, with each layer representing different qualities of the same video sequence [9] [10] [12].

The scalable extension of HEVC (SHVC) is very similar to multiview extension of HEVC. The compression efficiency in SHVC is achieved by inter-layer prediction and changes to the high-level syntax, without any changes to the block level coding tools of the single-layer HEVC standard. A simplified block diagram of scalable coding is depicted in Figure 3.7. As, shown the SHVC bitstream consists of two layers a Base layer (BL) and Enhancement layer (EL). The SHVC bitstream may consist of more than one ELs. The BL is backward compatible with single layer HEVC standard and is the lowest quality representation. The ELs may be coded by referring the BL or other lower ELs and they provide improved video quality.

Scalability is achieved in mainly three ways: temporal, spatial and quality scalability. The first two types of scalability correspond to sub-bitstreams, where, the source content is a reduced picture size and frame rate respectively. The quality scalability, also referred to as signal-to-noise ratio (SNR) scalability or fidelity scalability, refers to a sub-bitstream, where, the source content is of the same resolution and frame rate but represented with lower reproduction quality and thus lower bitrate. In case of spatial scalability, the BL is the down-sampled version of the EL. In case of quality

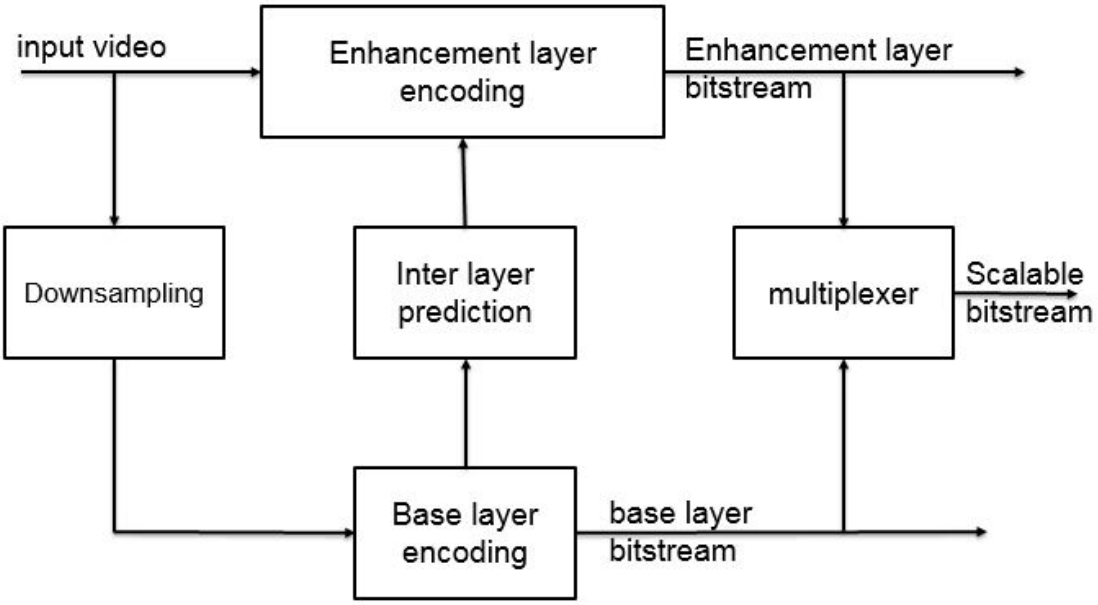


***Figure 3.7*** *A simplified block diagram of scalable encoder with two layers.*

scalability, the BL has the same input as the EL. The improvement of SHVC over simulcast streaming of video sequences come from inter-layer prediction methods. It uses data from the BL for efficient coding of the EL.

The SHVC standard allows two types of BL bitstream transmission. In the first case the BL bitstream is sent as part of the SHVC bitstream also known as in-band transmission. At the decoder the BL bitstream is de-multiplexed from the SHVC bitstream and decoded by the BL decoder. Efficiency in inter-layer prediction of EL is achieved by processing the reconstructed BL obtained from the decoded picture buffer of BL; using the processed BL as inter-layer reference in the decoded picture buffer of the EL. In the second case the BL stream is provided through external means, for example, other system level multiplexing methods. This functionality is provided by the SHVC mainly to support non-HEVC based BL bitstream, for example, with H.264/AVC single layer coding or other non-standardized codecs, this is also referred to as hybrid codec scalability. The BL bitstream provided through external means may also be compatible with HEVC coding standards and the decoding of BL bitstream is outside the scope of SHVC decoder, thus, there is no restriction on conformance of the BL bitreams provided externally. The decoded and reconstructed BL pictures are fed to the SHVC decoder along with information

associated with the BL pictures, the processing of EL is similar to the case one.

# 4. THE IMPLEMENTED VIDEO CODING ALGORITHMS

This chapter briefly describes, the methods implemented for the encoding of multi-view video content described in Chapter 2. The implemented methods are compliant with the video coding standards discussed in Chapter 3. Section 4.1 discusses the hierarchical GOP structure that is used in all the experimental methods. Sections 4.2 to 4.4 discusses the storage and streaming optimization methods as indicated in Table 4.1.

***Table 4.1*** *Experimented methods for storage and streaming bitrate optimization.*

| Methods | Description |
|---|---|
| **Simulcast coding** | Encoding videos sequences as separate bitstreams. Only temporal prediction used. |
| **Multiview coding (unconstrained)** | Encoding videos with inter-view prediction enabled at all frames. |
| **Multiview coding (constrained)** | Encoding videos with inter-view prediction enabled at only selected frames. |
| **Scalable coding (unconstrained)** | Scalable encoding of videos with inter-layer prediction enabled at all frames. |
| **Scalable coding (constrained)** | Scalable encoding of videos with inter-layer prediction enabled at only selected frames. |

## 4.1 The Hierarchical GOP Structure

In all the experimental methods to be discussed in upcoming sections, a hierarchical GOP structure is used for prediction and encoding. The hierarchical GOP structure is as shown in Figure 4.1. A temporal layer concept which provides temporal scalability has been used. Each frame is associated with a temporal level identifier $t_{id}$. With this concept, it is easy to extract a coded video sequence with lower temporal resolution from a given video sequence just by discarding all the NAL (Network Abstraction Layer) units with $t_{id}$ larger than a required value. Thus, this structure helps in situations, where, the network bandwidth is varying and the decoder capability is low [31].

## 4.2 Simulcast Coding

A simple method for encoding the videos is to code them as separate bitstreams or use frame packing. Figure 4.2 shows the GOP structure for simulcast coding of the

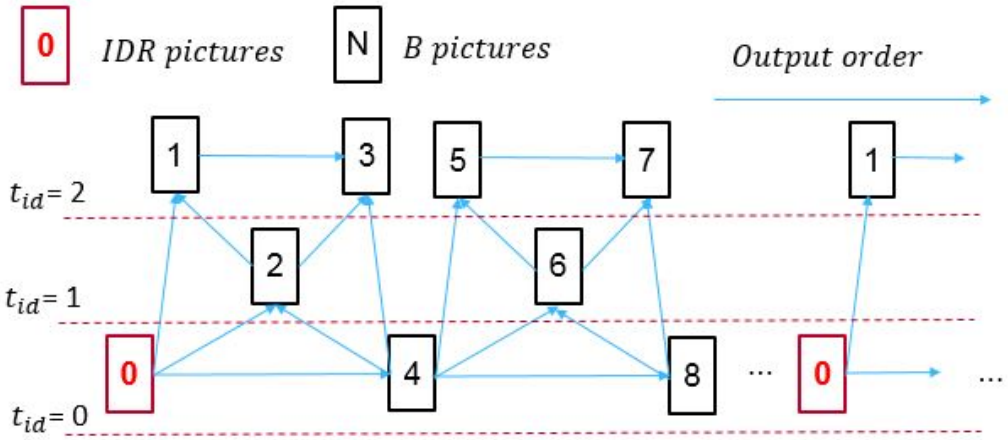


***Figure 4.1*** *The hierarchical GOP structure used for coding and prediction.*

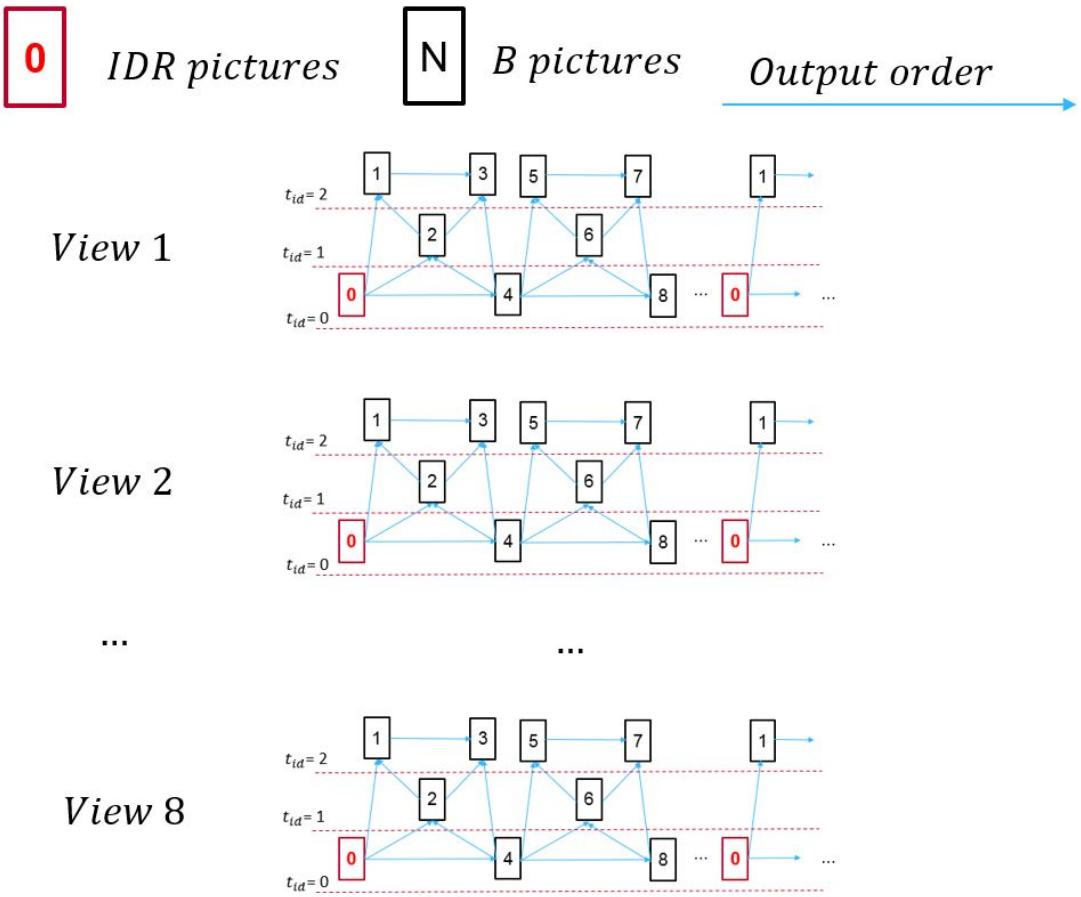


***Figure 4.2*** *The hierarchical GOP structure used in Simulcast coding of 8 views.*

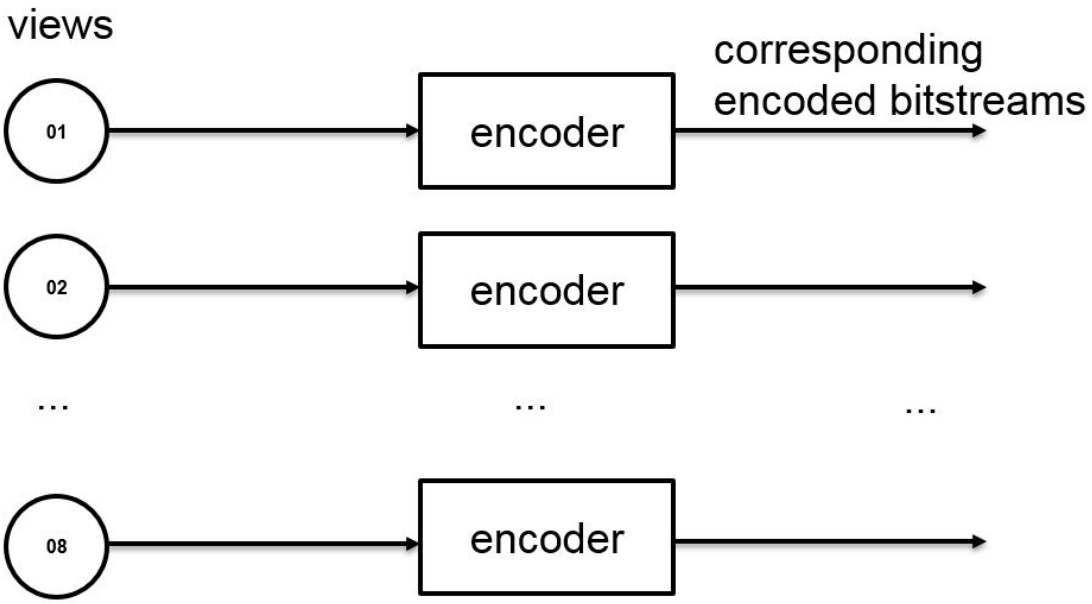


***Figure 4.3*** *Simulcast coding, encoding 8 views separately.*

8 camera views. It results in eight separate bistreams. The structure is equivalent to eight encoders coding parallely as shown in Figure 4.3. In the method of frame packing all the eight views are spatially packed into a single frame. This frame packing results in a single bitstream, with all the eight views available at the same time instance. However, the method of frame packing is not used in our analysis, as they produce video frames of very high resolution which may not be supported by off-the-shelf encoder. In this thesis, the simulcast method of coding is used for rate-distortion (RD) performance optimization.

## 4.3 Multiview Coding

The information in eight views are redundant. The method of simulcast and frame packed coding, discussed in section 4.2, reduces the redundancy temporally by using temporal prediction. However, it do not reduce the inter-view redundancy. These methods are suitable for video coding standards supporting single layer decoding and it keeps the encoding/decoding complexity at minimal. The presence of redundant data between views results in streams of higher bit-rates.

The following method minimizes the redundancy between views by making use of

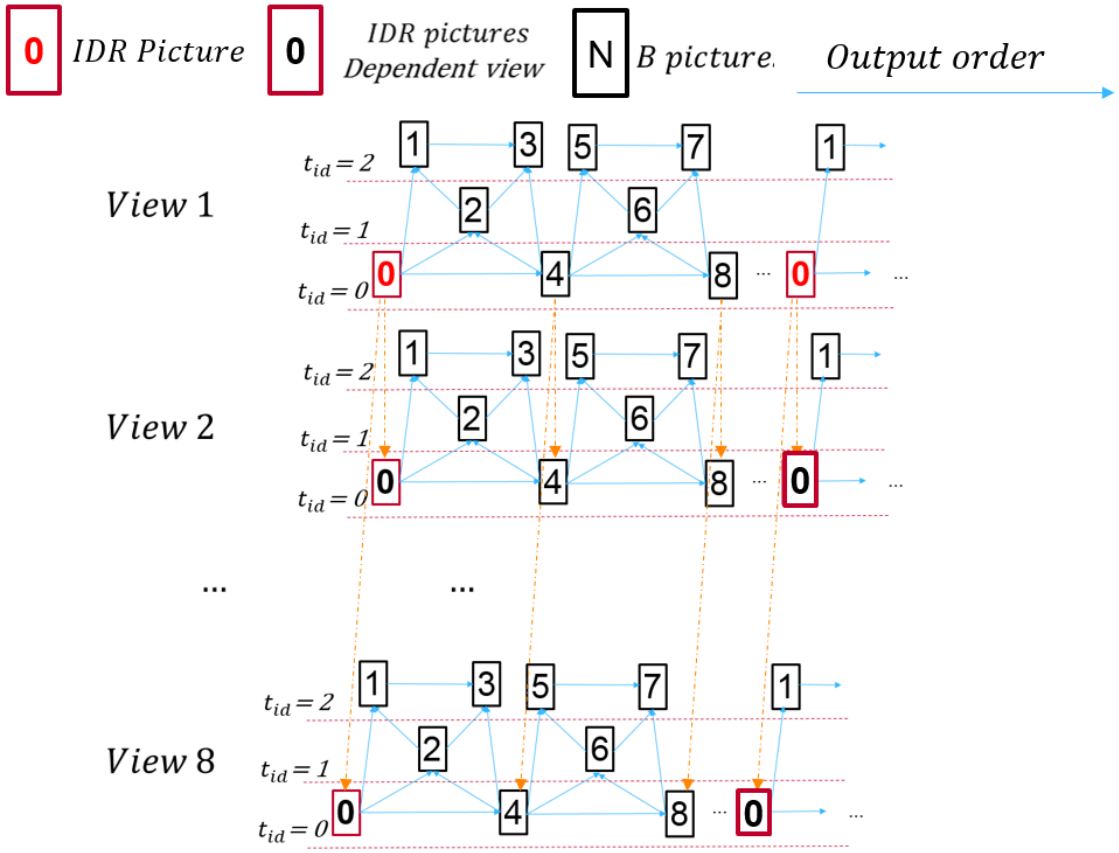


***Figure 4.4*** *The hierarchical GOP structure used in multiview prediction at every* $4^{th}$ *frame. In this example Views 2 to 8 are predicted from View 1.*

inter-view prediction. Two variants of the method were experimented. One in which inter-view prediction was enabled at every fourth frame of the GOP structure (constrained prediction). In the second case, the inter-view prediction was enabled in all the frames of the GOP structure (unconstrained prediction). The temporal scalability of the hierarchical GOP structure was retained in both the methods. The enabling of inter-view prediction in all frames increases the transmitted bitrate as both the base view and the dependent view is decoded at the display. However, this situation may not be optimal in cases when the current viewing direction is the dependent view or the user is restriced to view only a particular direction of the video. The inter-view prediction is not always efficient in coding of dependent frames. The benefits of inter-view prediction cannot balance the overhead bits, which need not be transmitted if there is no inter-view prediction. Hence, enabling inter-view prediction in selective frames improves the bitrate in these situations as it is not

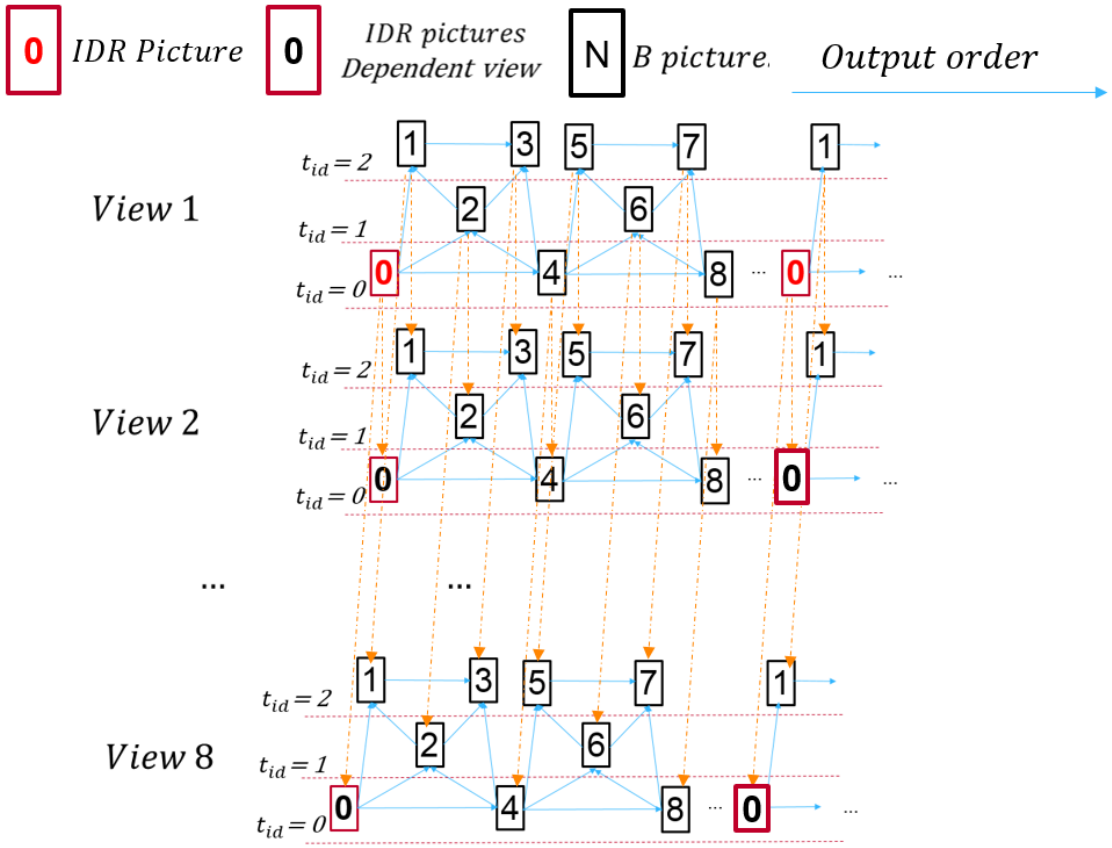


***Figure 4.5*** *The hierarchical GOP structure used multiview prediction in all frames. In this example Views 2 to 8 are predicted by View 1.*

required to transmit bits for the base view. The same argument can be carried over to the selective inter-layer prediction in section 4.4.

Figure 4.4 shows the inter-view prediction between views enabled at every fourth frame and Figure 4.5 shows the inter-view prediction between views enabled at all frames. As shown, views 1 and 4 are independent views and hence coded as base layers, where as, views 2,3, 5, 6, 7 and 8 are dependent views coded as P pictures from either view 1 or view 4 based on them being physical close to either view 1 or 4, respectively. The dependent views are not coded as B frames from both view 1 and 4 as it increase the overhead of transmitting base views in situations where it is not required due to current viewing direction of the user.

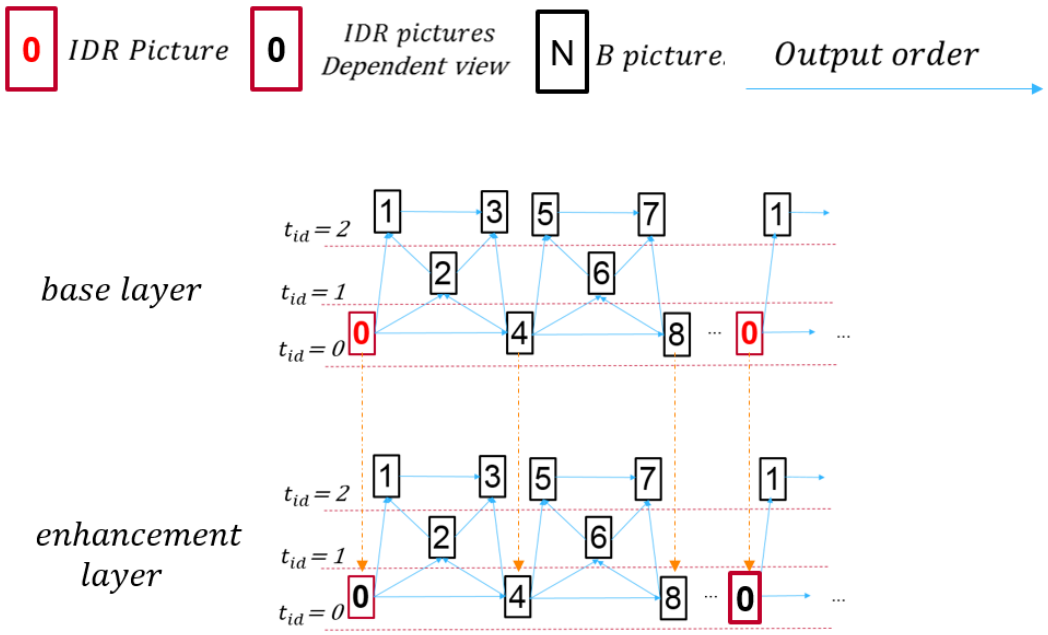


***Figure 4.6*** *The hierarchical GOP structure used in scalable coding. Inter-layer prediction is enabled at every* $4^{th}$ *frame.*

## 4.4 Scalable Coding

The methods of simulcast and multiview coding, encode videos with a given resolution and quality. However, these techniques do not address the following situations, such as, variations in the network bandwidth, videos streamed to mobile devices of heterogeneous capability, the current viewing direction of the end user. A good solution to these situations is the use of scalable coding tools, which enables the streaming of a source content in multiple quality and resolutions.

Figure 4.6 and Figure 4.7 shows the coding structure used in scalable coding of video sequences. It makes use of inter-layer prediction for efficient coding of the enhancement layer. Two variants of the inter-layer prediction have been experimented. In the first prediction structure of Figure 4.6, inter-layer prediction is used at every fourth frame. In Figure 4.7, the inter-layer prediction is used in all frames. The advantages of selective inter-view prediction has been discussed in Section 4.3, the same reasoning can be carried over to inter-layer prediction, as the tools of SHVC is similar to MV-HEVC, in both the tools video sequences are coded as multiple layers. In this thesis, views 1 and 4 are spatially scalable coded as they cover the

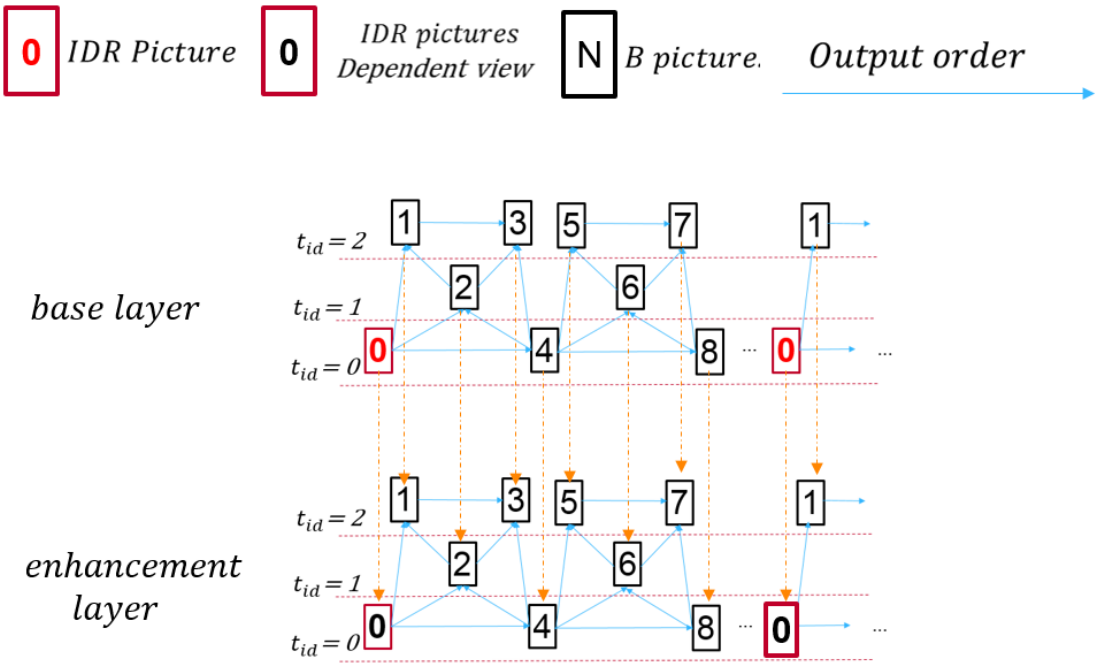


***Figure 4.7*** *The hierarchical GOP structure used in scalable coding. Inter-layer prediction is enabled at every frame.*

entire viewing world of the user and thus it helps in fast view switching. The reconstructed/decoded pictures are used for encoding other views in a multiview coding setup.

## 4.5 Designs for Streaming Bit-rate

The coding tools implemented in this section aim to optimize the streaming bitrate. The methods were designed based on the following assumptions: at any given time the end user views only a part of the $360^0$-degree world surrounding them. This leads to the opportunity that, it is not necessary to stream all the 8 views of the camera at all the time instants. The low latency requirement in responding to rapid view switching of the user makes it impossible to display the current stereoscopic view at highest quality immediately. Instead immediate monoscopic viewing can be guaranteed with cameras 1 and 4, as they cover the entire $360^0$-degree FOV, hence, are always streamed at a basic quality. This helps to keep the latency of display at minimal due to view switching. The other camera pairs are sufficient for stereoscopic viewing in a particular range of viewing directions in the "primary hemisphere"of

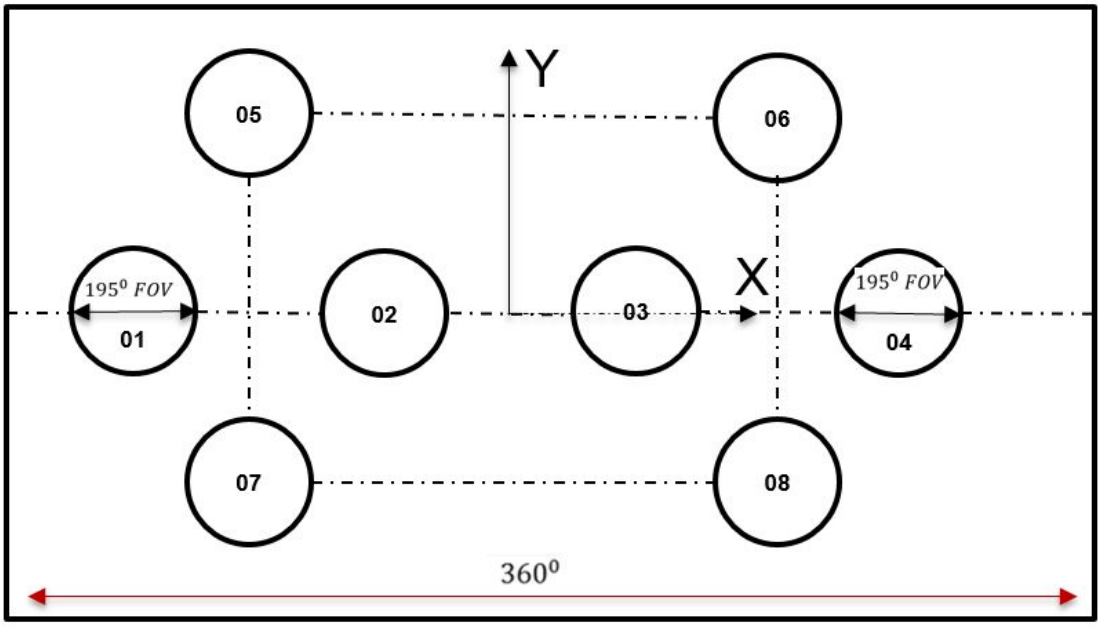


***Figure 4.8*** *The camera system projected on a rectangular grid.*

the camera. Thus, along with camera 1 and 4, other adjacent pair of cameras are used for displaying at the user end and hence is streamed at a better quality. These assumptions lead to 6 adjacent pair of cameras (1,2), (2,3), (3,4), (1,4), (5,6), (7,8) plus any cameras the coding of the pair depends on plus any coded representation of cameras 1 and 4 if not included in the streaming pair. The diagram depicting the camera system set-up is redrawn here in Figure 4.8.

Based on the above assumptions, 4 design techniques were implemented. The methods are discussed in the coming subsections

### 4.5.1 Simulcast Streaming

A simple method of streaming adjacent pairs of cameras is the simulcast coding. All the camera pairs are streamed at highest quality. This coding technique only utilizes the inter-frame prediction, temporally. All the proposed designs should improve upon the simulcast coding technique. Thus, this method is used as a reference for rate-distortion optimization of the proposed techniques.

An improvement in the streaming bitrate can be achieved simply by streaming the cameras 1 and 4 at a lower quality and all other adjacent pairs streamed at their

highest quality. This technique helps in reducing the latency of display at the user end due to fast view switching.

### 4.5.2 Multiview Streaming

This section describes the technique of multiview video coding for streaming adjacent camera pairs. It is proposed based on the camera system set-up. Either camera 1 or camera 4 are can be used as base views. The remaining adjacent views are predicted from the base views in a multiview set-up, this encoding scheme improves over simulcast coding by making use of inter-view prediction. Figure 4.9, shows the prediction of adjacent pairs from camera 1 and 4 based on the respective cameras being physically closer. Many other combinations of the prediction structure can also be used for coding. The same hierarchical GOP structure is used for coding, with inter-view prediction enabled at every $4^{th}$ or all frames based on the complexity requirements of the application.

### 4.5.3 Scalable + Multiview Streaming

This section discusses the concept of scalable + multiview coding applied for the streaming of camera pairs. Multiview representation of the $360^0$-degree content requires a large amount of data. Even with state-of-the-art, multiview coding tools the compression bitrates of multiview video is high. A combination of multiview and scalable coding was proposed by Kurutepe et.al in [32]. The method proposes to encode low resolution multiview videos in a multiview coding setup. The decoded and reconstructed multiviews are upsampled and are used as base layers to encode corresponding high resolution enhancement layers. This method improves the compression efficiency by having selective streaming of views basaed on the current viewing direction of the end user. It addresses the low latency requirement near the display by allowing random access to low resolution views in the base layer. However, this method does not improve the compression in situations where the multiview cameras are of high FOV as in Fisheye images as it requires to stream all the multiview videos in the base layer and hence cannot be used in the current thesis. Therefore the current thesis proposes a variant of the above method, where scalable coding is used in the base layer and multiview coding is used in the enhancement layer.

There are basically two methods discussed here. In both the methods, cameras 1 and 4 are scalable encoded with the base layer a down-sampled (by half) version

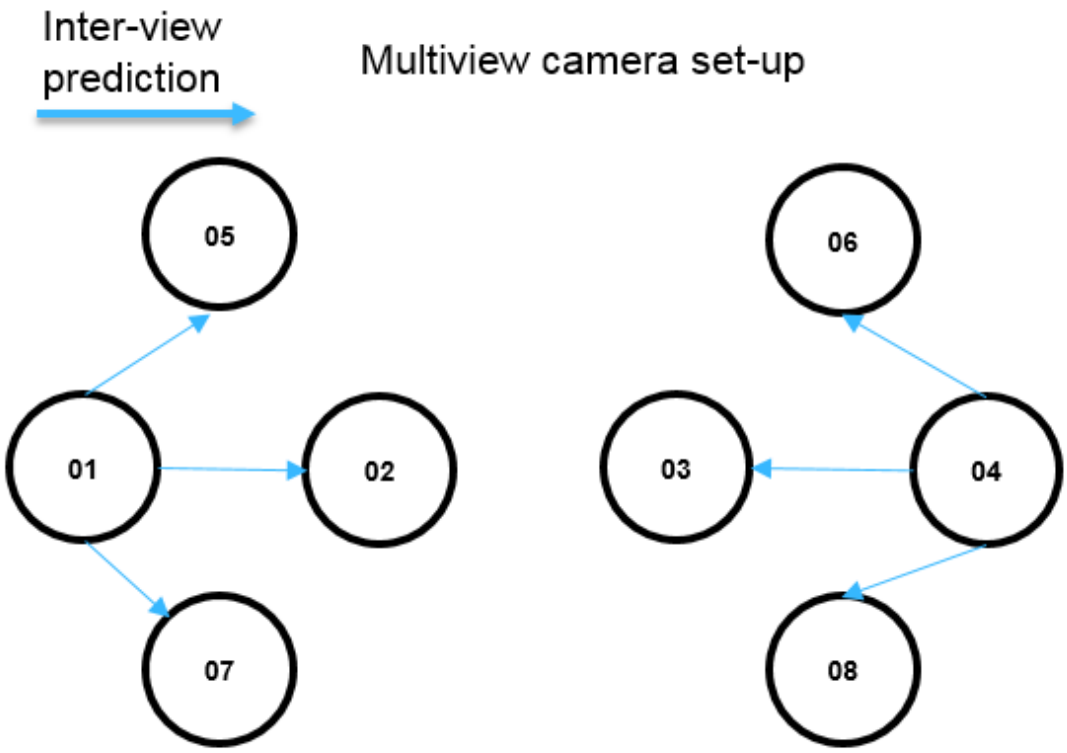


***Figure 4.9*** *The inter-view prediction structure for streaming adjacent pairs. In this example camera 4 and camera 1 is used as the base view.*

of the high quality enhancement layer and, the hierarchical GOP structure is used for coding and prediction. In the first method, high quality enhancement layers of cameras 1 and 4 is used as external base layer to encode other camera pairs. The enhancement layer bitstream of either camera 1 or camera 4 is used as the base layer to encode other cameras based on the respective cameras being physically closer.

Figure 4.10, shows the set-up that is used for coding adjacent camera pairs. The inter-layer prediction is enabled. Use of different views in the base and enhancement layers is same as multiview encoding with inter-view prediction. The example of Figure 4.10 uses the enhancement layer of camera 1 as base layer to encode camera 2 in the enhancement layer. The same set-up can be used to code views 5 and 7. It can also be extended to camera 4 as the base layer for coding views 3, 6 and 8 in the enhancement layer.

The second method is similar to the first method, but, the streaming bitrate is improved by now considering the decoded base layers of camera 1 and 4; their up-

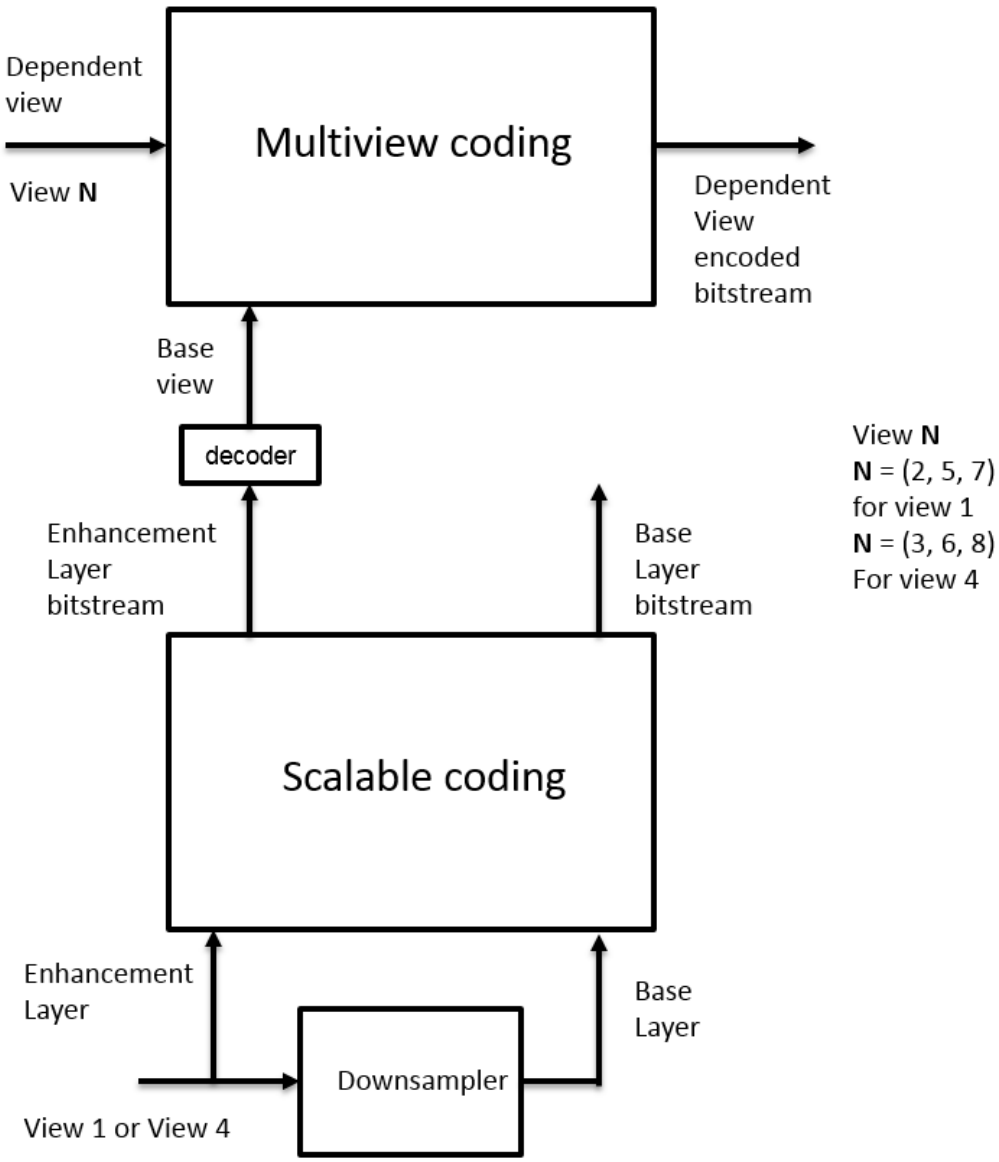


***Figure 4.10*** *Example of scalable coding with multiview coding scheme. Enhancement layer of camera 1 is used as the base layer to encode camera 2.*

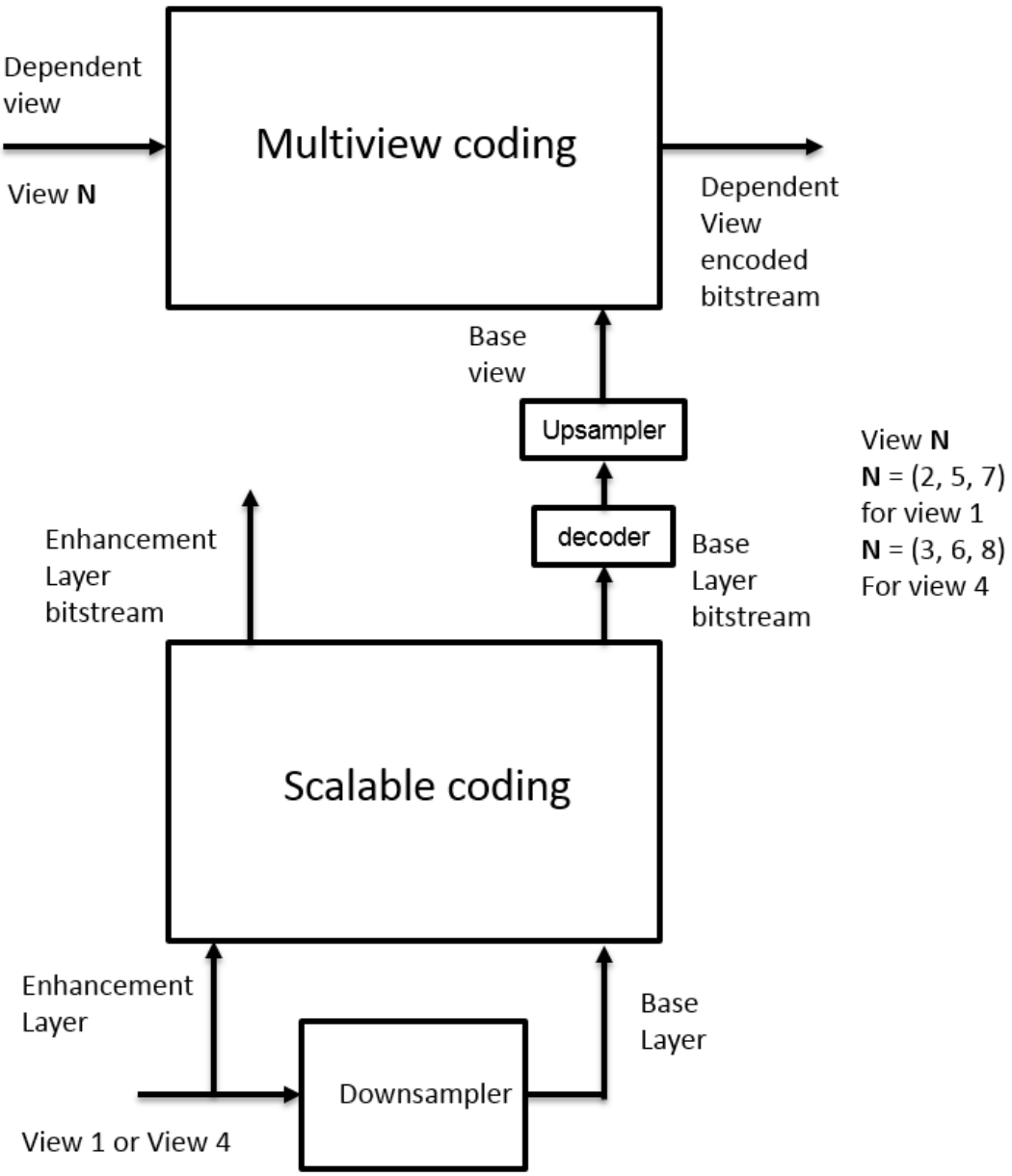


***Figure 4.11*** *Example of scalable skip coding with multiview coding scheme. Base layer of camera 1 is up-sampled and used as external base layer to encode camera 2.*

sampled (skip coded) version is used as base layers for coding the other cameras based on their physical closeness. Figure 4.11, shows an example where view 1 is scalable coded and the decoded base layer is up-sampled to be used for prediction of enhancement layers. The first method adds a constraint on the decoder and expects the network to provide the bandwidth for streaming high resolution videos even in the base layer as high resolution base layer is used for decoding of enhancement layers of the multiview bitstream. Whereas, the second method improves the compression efficiency by relaxing the constraints of the first method by allowing low resolution sequences of camera 1 and 4 in the base layer for decoding high resolution enhancement layer.

# 5. SIMULATION RESULTS

This chapter presents the results for the methods discussed in Chapter 4. Section 5.1, discusses the file format of video content and the metrics used for comparison of the obtained results. Section 5.2, presents the storage and streaming results of simulcast, multiview and scalable coding methods of Chapter 4. The last section presents the storage and streaming results for the methods which were proposed to optimize the streaming bitrate.

## 5.1 The Coding Framework

This section discusses the processing and evaluation steps used in the coding experiments.

### 5.1.1 Video Sequences

The videos used for experiment were captured from eight cameras which are set-up as discussed in Chapter 2. Two sets of multiview sequences were used in the encoding experiments. The video sequences are converted to YUV raw format for coding. The conversion was made with FFMPEG tools [33]. The two sequences are defined to be $SEQ\_SET1$ and $SEQ\_SET2$. The sequence $SEQ\_SET1$ has small object motion, while $SEQ\_SET2$ contain both object and camera motion. The videos are of 1408x1408 resolution and frame rate of 25 fps, 49 frames were used in encoding. Video frames from $SEQ\_SET1$ and $SEQ\_SET2$ are shown in Figure 5.1 and Figure 5.2, respectively.

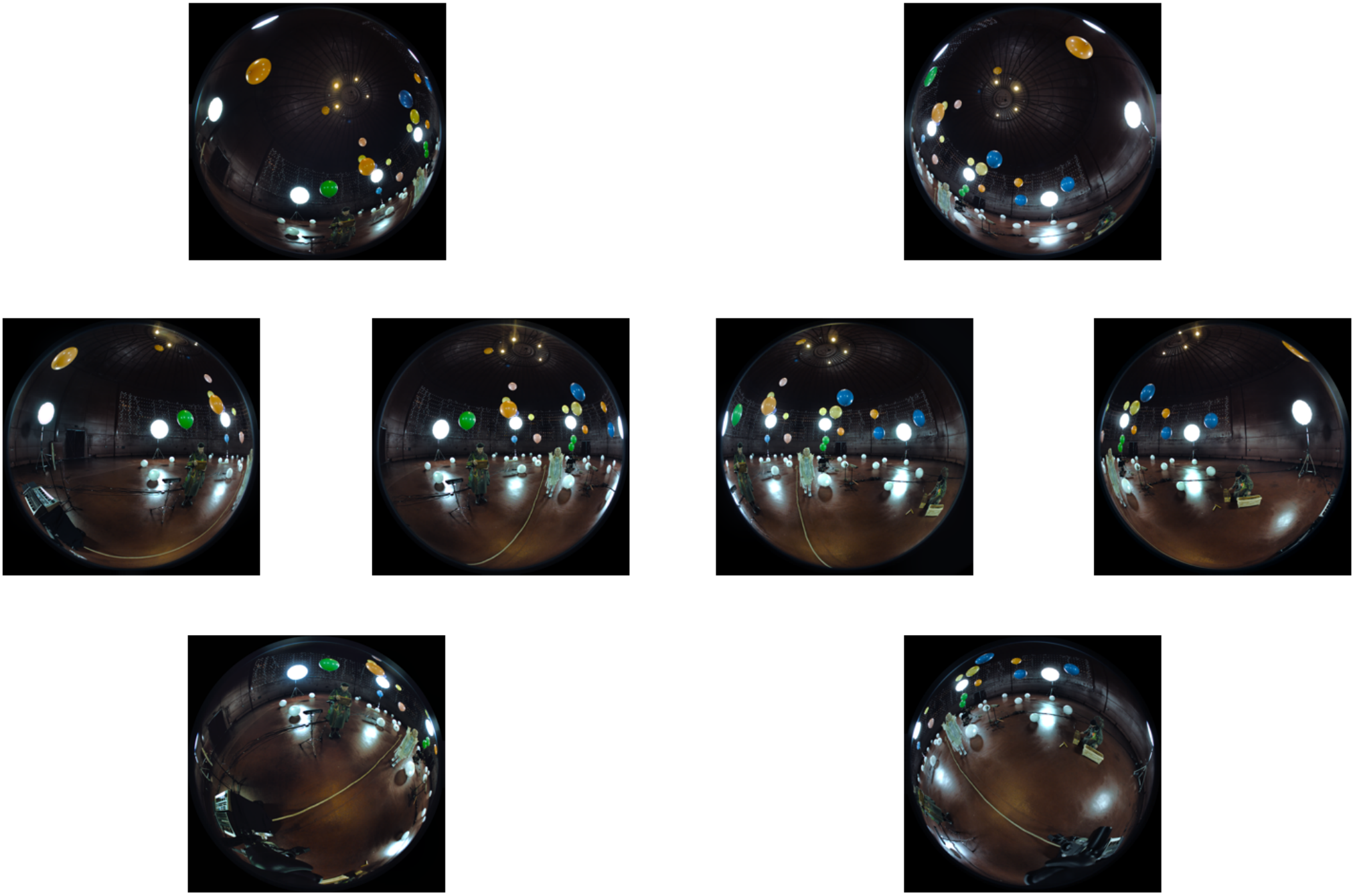

***Figure 5.1*** *Video frames from 8 cameras of the test sequence* $SEQ\_SET1$.

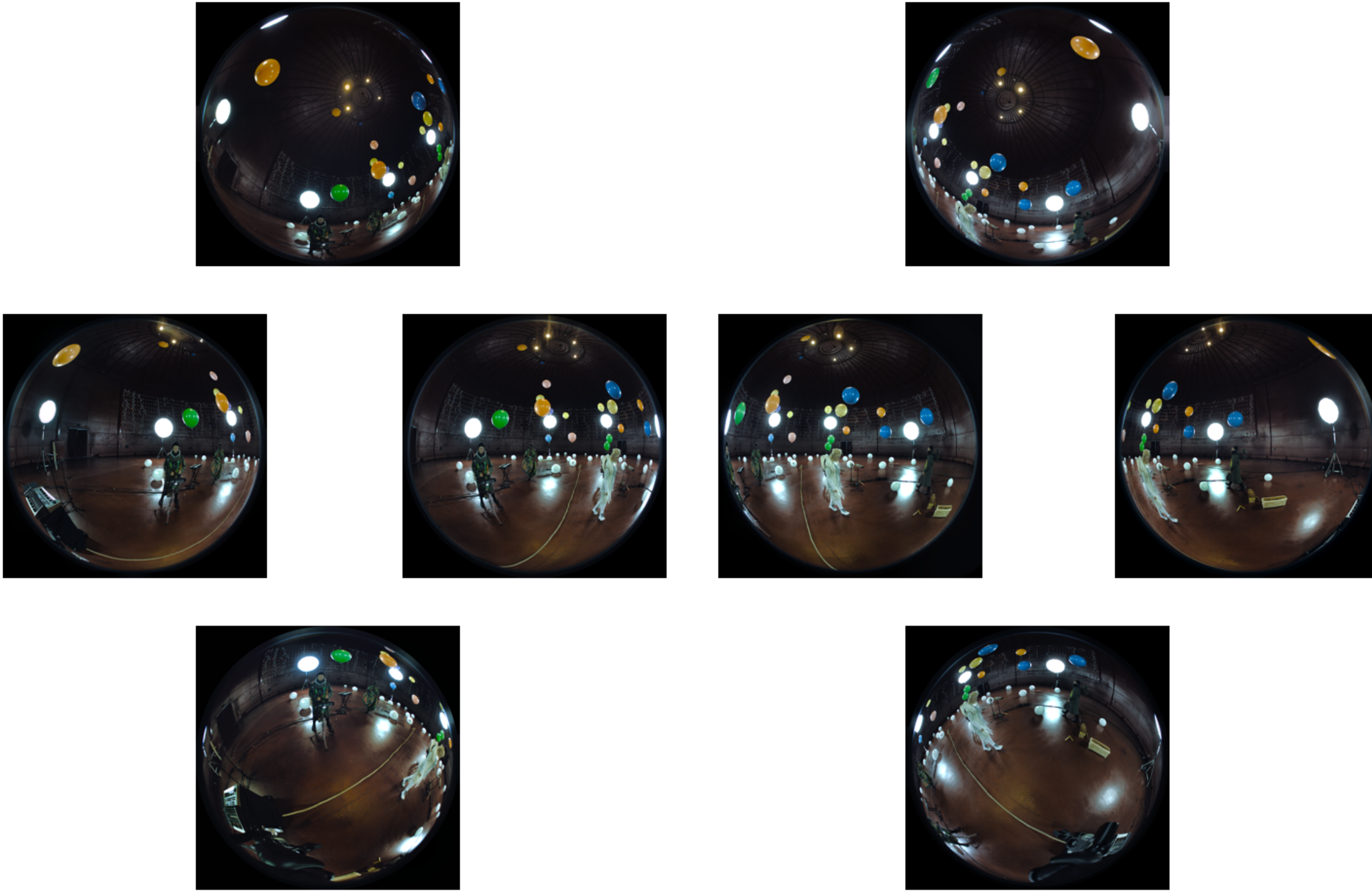

***Figure 5.2*** *Video frames from 8 cameras of the test sequence SEQ_SET2.*

### 5.1.2 Performance Metric

Many aspects are used for evaluation of video codecs. The general performance metrics used in the video coding community and the standards include bitrate (or compression ratio), computational cost (or complexity), quality (or distortion), scalability, error robustness, and interoperability.

The bitrate is measured in bits per second (bps or bits/s). Computational cost points to the processing power required for coding the video sequence. Quality is measured either subjectively or objectively. In this thesis work, objective quality is measured with PSNR (peak signal to noise ratio), in units of dB (decibels). The equation 5.1, shows the PSNR calculation used in image/video coding. The value 255 in equation 5.1 is the maximum value of the 8-bit luma samples. MSE is the mean square error given by the equation 5.2. The values N and M are the number of rows and columns, respectively in the video frame; $x_{ij}$ is the original pixel value at the position of $i_{th}$ row and $j_{th}$ column; $y_{ij}$ is the processed (such as decoded) pixel value at the position of $i_{th}$ row and $j_{th}$ column.

$$PSNR(dB) = 10 \cdot log_{10} \cdot \frac{(255)^2}{MSE} \tag{5.1}$$

$$MSE = \cdot \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (x_{ij} - y_{ij})^2}{NM} \tag{5.2}$$

### 5.1.3 Rate-Distortion Curve

The performance of different coding schemes is usually compared with rate-distortion curves. It is important to have a number representing the overall bitrate savings or the overall quality difference. In the current thesis work a delta calculation method proposed by Bjøntegaard has been used. This method has become the state-of-the-art evaluation metric in the context of video coding standardization [34] [35]. The two types of Bjøntegaard-Delta (BD) measurements available are used. The BD-rate provides a number for the overall rate savings in percent, and the BD-PSNR the overall PSNR difference. In all the coding methods, the videos are coded in 4 QP values. The QP values used in the experiments are 23, 28, 33 and 38.

### 5.1.4 Encoder Software

The following reference software of the video coding standards were used in all the encoding experiments. The JM version 18.0 reference software for H.264/AVC [5]. The HM version 16.0 reference software for H.265/HEVC [7]. The HTM version 14.0 reference software for the multiview extension of H.265/HEVC [36]. The SHM version 9.0 reference software for the scalable extension of H.265/HEVC [37].

## 5.2 Storage and Streaming Experiments

This section presents the results of encoding experiments on the video dataset $SEQ\_SET1$. The following methods were used for encoding.

1. **AVC Simulcast** - in this case, the 8 views are Simulcast coded with H.264/AVC encoder Main-profile. This method is used as a reference to measure the performance of HEVC simulcast coding.
2. **HEVC Simulcast** - the 8 views are simulcast coded with H.265/HEVC encoder Main-profile. The performance of all the remaining methods is compared against the HEVC simulcast coding.
3. **MV-HEVC Unconstrained** - the 8 views are coded with inter-view prediction at all frames. The multiview extension of HEVC is used for encoding. Cameras 1 and 4 are the base views and all the remaining camera views are predicted from either camera 1 or 4 based on their physical closeness.
4. **MV-HEVC Constrained** - this is similar to MV-HEVC Unconstrained, with inter-view prediction enabled at only every $4^{th}$ frame.
5. **SHVC Simulcast** - in this method, all the 8 views are scalable coded. The base layer is the down-sampled (by half) version of the enhancement layer. Videos from the same view are used in both base and enhancement layers.
6. **SHVC + MV-HEVC Constrained** - this method combines the tools of SHVC and MV-HEVC. Cameras 1 and 4 are scalable coded with SHVC. All the remaining cameras are multiview coded either using camera 1 or camera 4 as base layers. The inter-layer prediction is enabled at every $4^{th}$ frame.

7. **SHVC + MV-HEVC Unconstrained** - this is the same method as SHVC + MV-HEVC Constrained but, with inter-layer prediction enabled at all frames.

Table 5.1, shows the Bjøntegaard rate-distortion results. The corresponding BD-curve is shown in Figure 5.3. AVC Simulcast is compared against HEVC Simulcast. All the remaining methods are compared against HEVC Simulcast. The results show that HEVC improves over AVC by approximately 34%.

***Table 5.1*** *Storage bitrate for the 7 Methods in storage and streaming experiments.*

| **Methods** | **Bjøntegaard results** | |
|---|---|---|
| | **dBR** | **dPSNR (dB)** |
| **AVC Simulcast** | **reference** | **reference** |
| **HEVC Simulcast** | **-34.71%** | **1.16%** |
| **HEVC Simulcast** | **reference** | **reference** |
| **MV-HEVC Unconstrained** | **-2.03%** | **0.05%** |
| **MV-HEVC Constrained** | **-1.92%** | **0.00%** |
| **SHVC Simulcast** | **13.52%** | **-0.37%** |
| **SHVC + MV-HEVC Constrained** | **-5.36%** | **0.13%** |
| **SHVC + MV-HEVC Unconstrained** | **-7.92%** | **0.25%** |

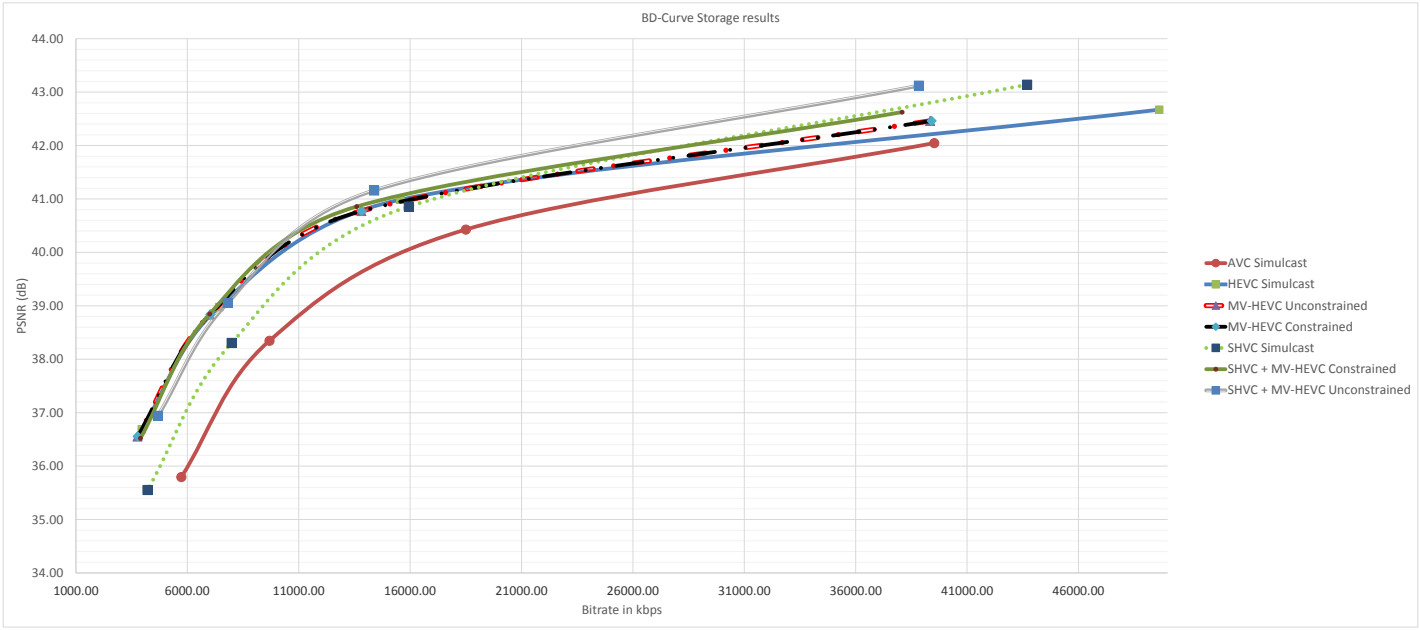


***Figure 5.3*** *BD curve for storage results of Table 5.1.*

**Streaming Bitrate** - the streaming results are calculated based on the following: Cameras 1 and 4 cover the entire $360^0$-degree FOV, hence, are always streamed at a basic resolution. Along with these two cameras the other adjacent pair of cameras, used for displaying at the user end is streamed at a better quality. These assumptions lead to 6 adjacent pair of cameras (1,2), (2,3), (3,4), (1,4), (5,6), (7,8) plus any cameras the coding of the pair depends on plus any coded representation of cameras 1 and 4 if not included in the streaming pair.

The Table 5.2, shows the rate-distortion values for the streaming of adjacent pairs in Method 1 to 7. The corresponding BD-curve is shown in Figure 5.4.

***Table 5.2*** *Streaming bitrate for the Methods in storage and streaming experiments.*

| **Methods** | **Bjøntegaard results** | |
|---|---|---|
| | **dBR** | **dPSNR (dB)** |
| **AVC Simulcast** | **reference** | **reference** |
| **HEVC Simulcast** | **-26.93%** | **0.85%** |
| **HEVC Simulcast** | **reference** | **reference** |
| **MV-HEVC Unconstrained** | **-2.40%** | **0.06%** |
| **MV-HEVC Constrained** | **-2.19%** | **-0.01%** |
| **SHVC Simulcast** | **9.06%** | **-0.23%** |
| **SHVC + MV-HEVC Constrained** | **10.77%** | **-0.28%** |
| **SHVC + MV-HEVC Unconstrained** | **8.68%** | **-0.32%** |

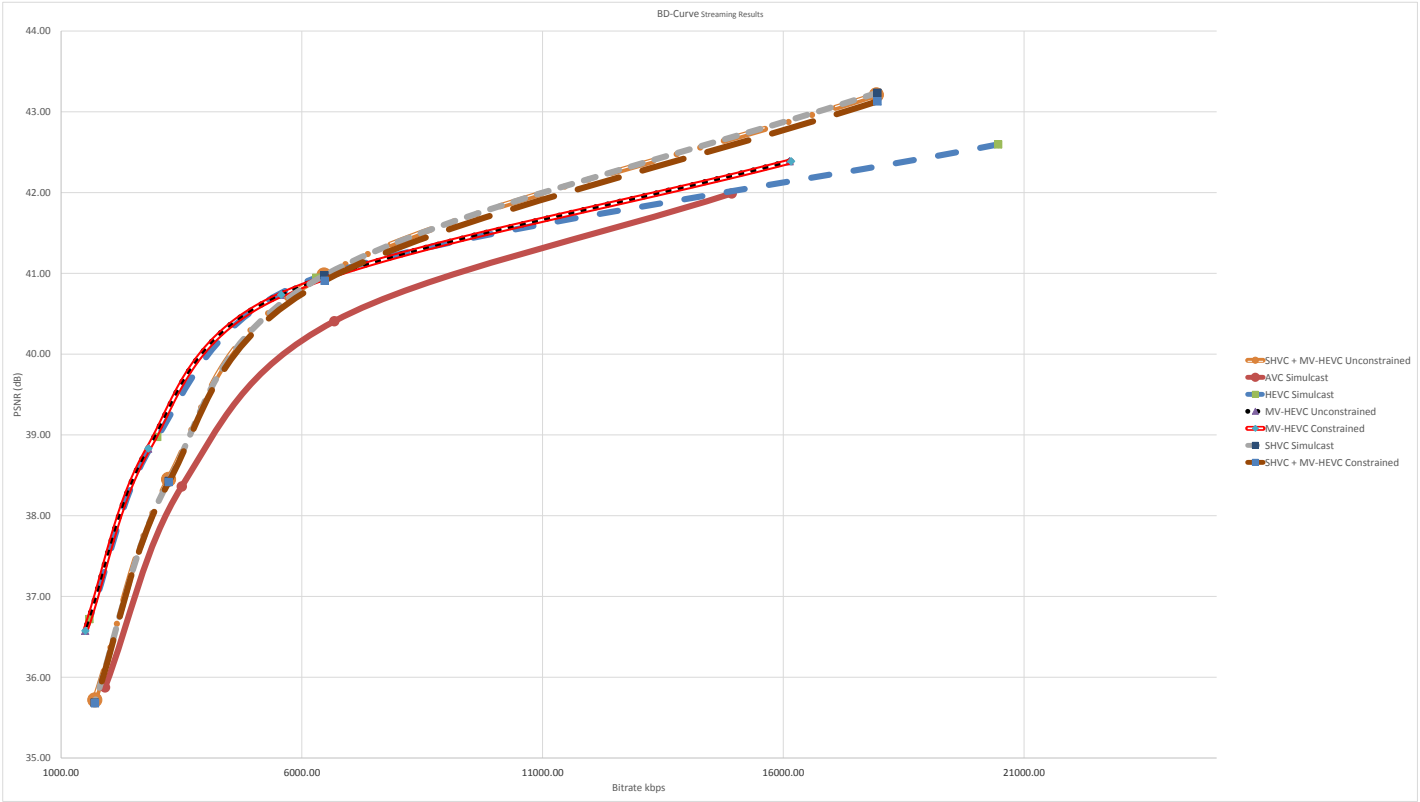


***Figure 5.4*** *BD curve for streaming results of Table 5.2.*

## 5.3 Streaming Optimization Experiments

This section presents the results of encoding experiments in optimizing streaming bitrate, on the video datasets $SEQ\_SET1$ and $SEQ\_SET2$. The following methods were used for encoding.

1. **HEVC Simulcast** - in this case the 8 views are Simulcast coded with H.265/HEVC encoder. Only a pair of views is streamed along with cameras 1 and 4 with highest resolution. This method is used as a reference to compare the performance of the remaining methods.
2. **MV-HEVC Constrained** - this method uses the multiview extension of HEVC for coding. Similar to the HEVC Simulcast, only a pair of views is streamed at a given time. However, these views are now predicted from either camera 1 or 4. Inter-view prediction is enabled at every $4^{th}$ frame.
3. **HEVC Simulcast Mixed resolution** - this is a simple simulcast method where a given pair of views is streamed in high quality, while cameras 1 and 4 are streamed in lower quality. The camera pairs are not predicted by any of camera 1 or 4.
4. **SHVC + MV-HEVC Constrained** - in this case cameras 1 and 4 are scalable coded with the base layer a down-sampled version of the high quality enhancement layer. The high quality enhancement layers of cameras 1 and 4 are used as external base layers to encode other camera pairs.
5. **SHVC + MV-HEVC Constrained Skip coded** - this is same as Streaming Method 4. However, in this method the decoded base layers of camera 1 and 4 are up-sampled and used as external base layers for coding the other cameras.

The Table 5.3, shows the Bjøntegaard rate-distortion values results. The corresponding BD-curve is shown in Figure 5.5.

The Table 5.4, shows the Bjøntegaard rate-distortion results for the streaming of adjacent pairs in the respective methods The corresponding BD-curve is shown in Figure 5.6.

***Table 5.3*** *Storage bitrate for the Methods in streaming optimization experiments.*

| Methods | Bjøntegaard results | |
|---|---|---|
| | dBR | dPSNR (dB) |
| HEVC Simulcast | reference | reference |
| MV-HEVC Constrained | -2.74% | 0.07% |
| HEVC Simulcast Mixed resolution | 2.81% | 0.03% |
| SHVC + MV-HEVC Constrained | -4.92% | 0.11% |
| SHVC + MV-HEVC Constrained Skip coded | -4.61% | 0.10% |

***Table 5.4*** *Streaming bitrate for the Methods in streaming optimization experiments.*

| Methods | Bjøntegaard results | |
|---|---|---|
| | dBR | dPSNR (dB) |
| HEVC Simulcast | reference | reference |
| MV-HEVC Constrained | -2.94% | 0.07% |
| HEVC Simulcast Mixed resolution | -25.50% | 0.71% |
| SHVC + MV-HEVC Constrained | -19.71% | 0.55% |
| SHVC + MV-HEVC Constrained Skip coded | -36.00% | 1.20% |

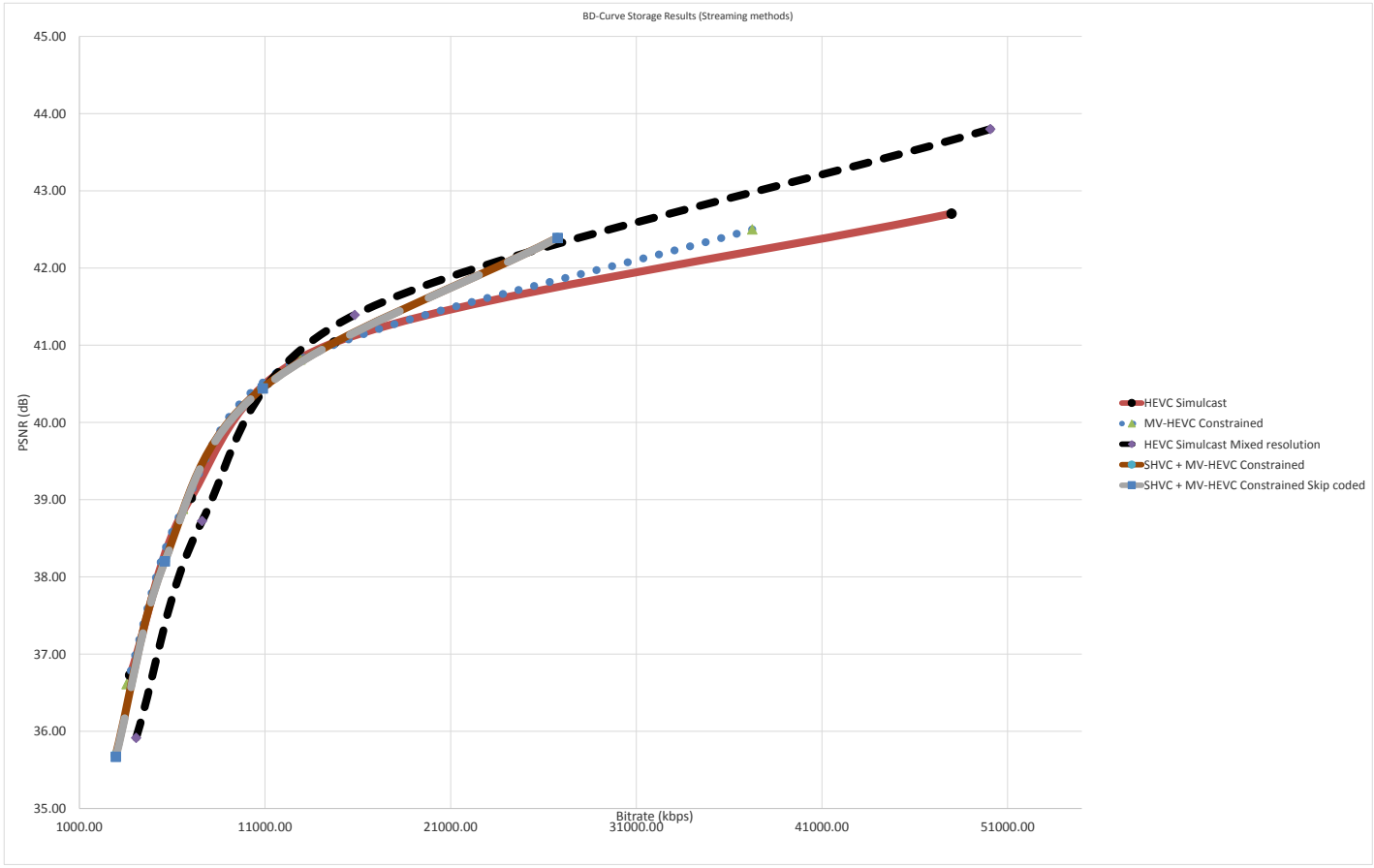


***Figure 5.5*** *BD curve for storage results of Table 5.3.*

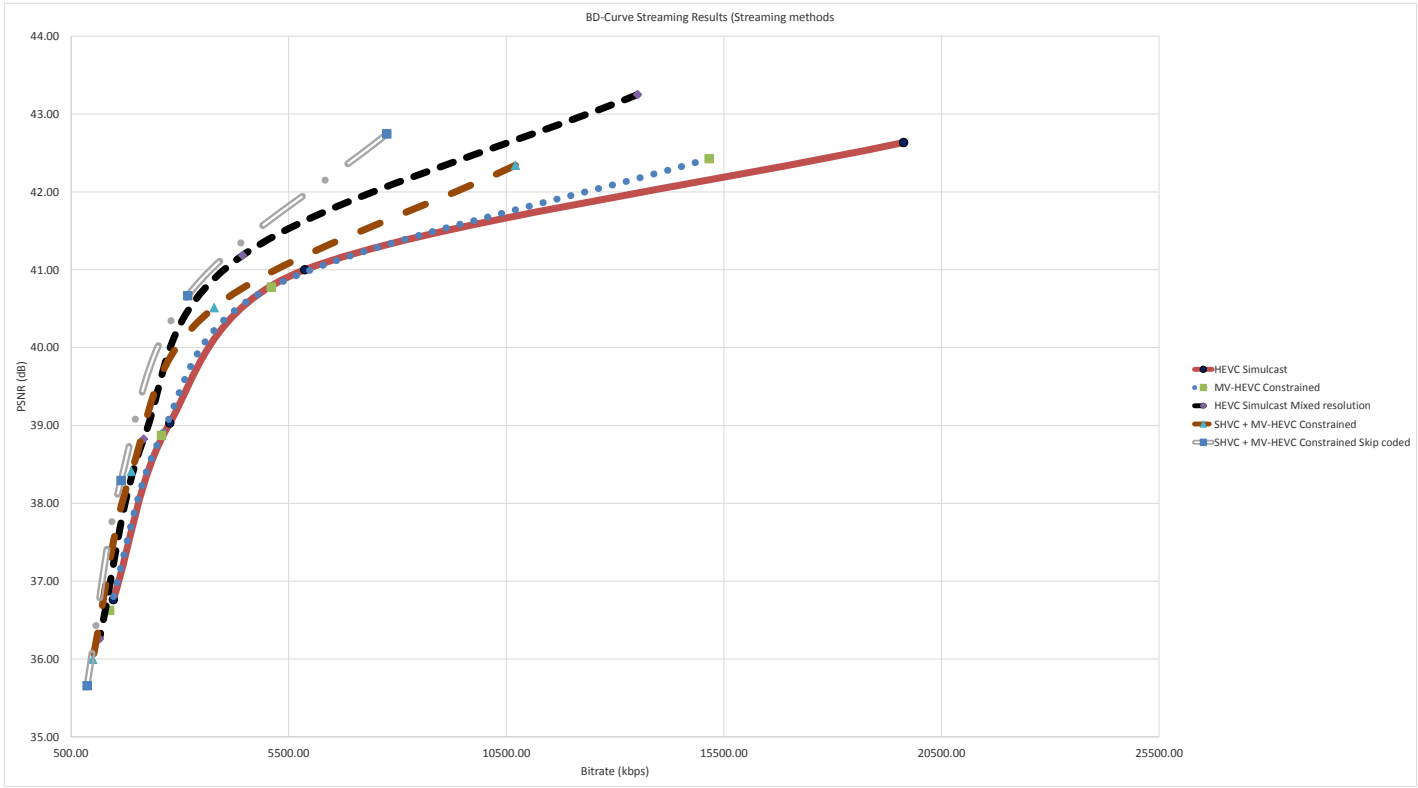


***Figure 5.6*** *The BD curve for streaming results of Table 5.4.*

In the first category of minimizing storage bitrate, the methods of simulcast coding, scalable coding of individual views and multiview and scalable coding based on camera 1 and camera 4 were experimented. In the case of last two methods, inter-view/inter-layer prediction was enabled at either every $4^{th}$ frame or in all frames. The method of simulcast coding of all 8 views was compared between H.264/AVC and H.265/HEVC. The HEVC method gave a bitrate reduction of approximately 34%, due to its improved coding tools. The scalable coding scheme, based on either camera 1 or camera 4 in the base layer and with inter-layer prediction enabled at all frames, gave the best bitrate reduction for storage of all the 8 views. The scalable coding is a good solution for heterogeneous devices as it helps in switching to a lower spatial quality based on the decoder capacity. However, it brings an overhead in streaming.

In the second set of solutions for optimizing streaming bitrates, five types of coding schemes were experimented. The simulcast coding of 8 views with H.265/HEVC was used as reference for comparing the proposed methods. In this case the scalable coding scheme with the views predicted from either camera 1 or camera 4 gave the best bitrate reduction in both storage and streaming. The best method for streaming was the scalable skip coded views, with an improvement of 36% on average over the two data sets. The results from two categories show that hierarchical GOP structures along with scalable coding schemes are the best solution for the VR applications as it helps in addressing the variations in network bandwidth, decoding capability and system latency by allowing temporal and spatial switching between views at the user display.

# 6. CONCLUSIONS AND FUTURE WORK

Virtual reality medium gives a sense of real world experience using videos in simulated environments. Videos in VR system are captured from multiple view points to cover the entire three-dimensional space of the world. The aim of the thesis was to make an extensive study of the existing video coding standards and propose coding schemes for VR systems. The coding schemes were implemented based on the following factors: constraints influencing the storage and streaming of video sequences in the VR system. Secondly, the designs should be complex enough to efficiently compress the multiview video content and yet simple enough, so that the encoding schemes can be employed in practical applications. The coding schemes proposed were experimented with multiview video sequences, which was captured from a spherical camera set-up. Eight cameras produced circular fisheye videos, with each covering $195^0$ degree FOV.

In this study, the video coding standards of H.264/AVC and especially its successors H.265/HEVC and multiview/scalable coding extensions have been investigated. These standards are the current state-of-the-art and have been recognized widely in the multimedia industry. The designs of video coding schemes were also influenced by other factors, such as, varying network bandwidth, heterogeneous mobile devices, the current viewing direction of the user, lower latency for faster view switching in the display.

For prediction and coding, a hierarchical group of picture structure was used. This coding structure provides temporal scalability and helps in switching between different temporal resolutions. It is an efficient solution to varying network bandwidth and decoders with low decoding capabilities. As the GOP size is small it also allows for efficient random access between views and reduces the overall latency of the system due to fast view switching. The same prediction and coding structure was carried over to all the proposed coding schemes. The design was mainly categorized into two types, one, which minimized the storage bitrate, second, which optimized

the streaming bitrate. However, both storage and streaming bitrates were calculated for all the methods in order to analyse the overhead at both the server and decoder end. The proposed coding schemes competed against the simulcast coding of individual views. The HEVC method gave a bitrate reduction of approximately 34%, as compared to AVC simulcast, due to its improved coding tools. The scalable plus multiview coding scheme, based on either camera 1 or camera 4 in the base layer and with inter-layer prediction enabled at all frames, gave the best bitrate reduction for storage of all the 8 views. The scalable plus multiview coding is a good solution for heterogeneous devices as it helps in switching to a lower spatial quality based on the decoder capacity. However, it brings an overhead in streaming, due to the requirement of streaming base views in all frames. The best method for streaming was the scalable plus multiview skip coding scheme, which an improvement of 36% on average against the HEVC simulcast method . The results from two categories show that hierarchical GOP structures along with scalable plus multiview coding schemes are the best solution for the VR applications as they helps in addressing the variations in network bandwidth, decoding capability and system latency by allowing temporal and view switching at the user display.

Even though the proposed methods of the thesis are confined by the video coding standards, some of the methods should be further developed for practical implementations. For examples, in the methods utilizing data across layers/views, there is a need to develop faster algorithms which operate in parallel in-order decrease the system latency. Furthermore many of the coding schemes should be tested in comprehensive applications and system environments. At last, subjective results of the schemes should be evaluated against the implementation cost and requirements of the processing power.

The thesis concentrated on minimizing the bitrate of multiview data set in a VR system, leaving out certain aspects, such as, fisheye distortion, pre-processing, outside of the scope of the thesis. For example, pre-processing the fisheye videos to rectilinear projection may help further, as rectilinear videos fit to the translational motion model of the video standards. Furthermore tile-based video coding may help in fast view switching, with the tiles coded based on the FOV of the user displays. Thus, potential future work could provide more analysis on aspects beyond bitrate reduction.

# BIBLIOGRAPHY

[1] A. B. Craig, W.R. Sherman and J.D. Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Elsevier, 2009.

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

[3] C.Mueller, S.Lederer, B.Rainer, M.Waltl, M.Grafl, and C. Timmerer, "Open Source Column: Dynamic Adaptive Streaming over HTTP Toolset", Alpen-Adria-Universität Klagenfurt, Institute of Information Technology, Multimedia Communication, ACM SIGMM Records, Vol. 5, No. 1, March 2013.

[4] ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services, International Telecommunication Union (ITU-T), May 2003.

[5] H.264/AVC reference software, Available at: `http://iphome.hhi.de/suehring/tml/` (March-02-2016).

[6] ITU-T Recommendation H.265, High efficiency video coding, International Telecommunication Union (ITU-T), April 2015.

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

[8] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services,"EURASIP Journal on Advances in Signal Processing, vol. 2009.

[9] M. M. Hannuksela, Y. Yan, X. Huang, and H. Li, "Overview of the multiview high efficiency video coding (MV-HEVC) standard", IEEE International Conference on Image Processing, pp. 2154-2158, September 2015.

[10] G. Tech, Y. Chen, K. Muller, J. R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of High Efficiency Video Coding,"IEEE Transactions Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 35-49, January 2016.

[11] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Transactions Circuits and Systems for Video Technology, VOL. 17, NO. 9, September 2007.

[12] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of SHVC: scalable extensions of the High Efficiency Video Coding standard", IEEE Transactions Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 20-34, January 2016.

[13] Project site on VP9 standard, Available at: `http://www.webmproject.org/vp9//` (March-02-2016).

[14] VP9 Overview and progress Update, Available at: `http://downloads.webmproject.org/ngov2012/pdf/04-ngov-project-update.pdf` (March-02-2016).

[15] A. Fuldseth, G. Bjontegaard, M. Zanaty, "Thor Video Codec draft-fuldseth-netvc-thor-00", Cisco Systems, July 6, 2015, Available at: `https://tools.ietf.org/html/draft-fuldseth-netvc-thor-00/` (March-02-2016).

[16] Project site on Daala standard, Available at: `https://xiph.org/daala/` (March-02-2016).

[17] Oculus notes on VR video recommendations, Available at: `https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_app_rendering/` (January-21-2016).

[18] R. W. Wood, "Physical Optics", pp. 66-68, The Macmillan Company, New York (1911).

[19] A. A. Muhit, M. R. Pickering, M. R. Frater, J. F. Arnold, "Video coding using fast geometry-adaptive partitioning and an elastic motion model", Elsevier, Journal of Visual Communication and Image Representation, Volume 23, Issue 1, January 2012.

[20] I. E. G. Richardson, "H.264 and MPEG-4 Video Compression, Video Coding for Next-generation Multimedia", John Wiley and Sons Ltd, 2003.

[21] M. Wien, "High Efficiency Video Coding, Coding Tools and Specification", Springer 2015.

[22] Axis communication White paper, "H.264 video compression standard. New possibilities within video surveillance".

[23] Joint Video Team (JVT), ITU-T website, Available at: `http://www.itu.int/ITU-T/studygroups/com16/jvt/` (March-02-2016).

[24] K. R. Rao, D. N. Kim, J J. Hwang, "Video Coding Standards, AVS China, H.264/MPEG-4 PART 10, HEVC, VP6, DIRAC and VC-1", Springer 2014.

[25] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards Including High Efficiency Video Coding (HEVC),"IEEE Transactions Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1669-1684, 2012.

[26] G. J. Sullivan, Fellow, IEEE, J.R. Ohm, Member, IEEE, W.J. Han, Member, IEEE, and T. Wiegand, Fellow, IEEE, "Overview of the High Efficiency Video Coding, (HEVC) Standard", IEEE Transactions Circuits and Systems for Video Technology, VOL. 22, NO. 12, December 2012.

[27] 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.

[28] J. Lainema, F. Bossen, Member, IEEE, W.J. Han, Member, IEEE, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard", IEEE Transactions Circuits and Systems for Video Technology, VOL. 22, NO. 12, December 2012.

[29] A. Vetro, D. Tian, "Analysis of 3D and Multiview Extensions of the Emerging HEVC Standard", SPIE Applications of Digital Image Processing, 2012.

[30] A. Vetro, Y. Chen and K. Mueller, "HEVC-Compatible Extensions for Advanced Coding of 3D and Multiview Video", Data Compression Conference (DCC), 2015.

[31] R. Sjöberg, Y. Chen, A. Fujibayashi, M. M. Hannuksela, J. Samuelsson, T. K. Tan, Y.-K. Wang, and S. Wenger, "Overview of HEVC high-level syntax and reference picture management,"IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1858-1870, Dec. 2012.

[32] 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.

[33] FFMPEG framework website, Available at: https://www.ffmpeg.org/ (March-14-2016).

[34] G.Bjøntegaard, "Calculation of average PSNR differences between RD curves", VCEGM33. ITU-T SG16/Q6 VCEG, Austin, USA, 2001.

[35] G.Bjøntegaard, "Improvements of the BD-PSNR model", VCEG-AI11. 35th meeting: ITU-T SG16/Q6 VCEG, Berlin, 2008.

[36] Multiview extension of H.265/HEVC reference software, Available at: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/trunk/ (March-02-2016).

[37] Scalable extension of H.265/HEVC reference software, Available at: https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/trunk/ (March-02-2016).