UNIVERSITY OF TARTU

FACULTY OF SCIENCE AND TECHNOLOGY

Institute of Technology

Robotics and Computer Engineering

Suman Sarkar

Dynamic Rate Allocation of Interactive Multi-View Video with View-Switch Prediction

Master's Thesis (30 ECTS)

Supervisors: Assoc. Prof. Gholamreza Anbarjafari
Dr. Cagri Ozcinar

Tartu 2016

Dynamic Rate Allocation of Interactive Multi-View Video with View-Switch Prediction

Abstract:

In Interactive Multi-View Video (IMVV), the video has been captured by numbers of cameras positioned in array and transmitted those camera views to users. The user can interact with the transmitted video content by choosing viewpoints (views from different cameras in the array) with the expectation of minimum transmission delay while changing among various views. View switching delay is one of the primary concern that is dealt in this thesis work, where the contribution is to minimize the transmission delay of new view switch frame through a novel process of selection of the predicted view and compression considering the transmission efficiency. Mainly considered a real-time IMVV streaming, and the view switch is mapped as discrete Markov chain, where the transition probability is derived using Zipf distribution, which provides information regarding view switch prediction. To eliminate Round-Trip Time (RTT) transmission delay, Quantization Parameters (QP) are adaptively allocated to the remaining redundant transmitted frames to maintain view switching time minimum, trading off with the quality of the video till RTT time-span. The experimental results of the proposed method show superior performance on PSNR and view switching delay for better viewing quality over the existing methods.

CERCS codes: Imaging, image processing (T111)

Keywords: Interactive multi-view video, Zipf distribution, Hidden Markov Model, Round-Trip Time.

Dünaamiline Kiiruse Jaotamine Interaktiivses Mitmevaatelises Video Vaatevahetuse Ennustamineses

Lühikokkuvõte

Interaktiivses mitmevaatelises videos (IMVV) on video filmitud mitme jadas oleva kaamera poolt ning edastatakse need vaated kasutajatele. Kasutajad saavad valida erinevate vaatepunktide (jadas asuvate kaamerate videovoogude) vahel, oodates minimaalset viivitust vahetuste ajal. Vaadete vahetamisel tekkiv viivitus ongi selle magistriöö üks peamisi uurimisobjekte, kus panuseks on viivituse minimeerimine läbi uudse protsessi, mis ennustab järgmisena valitava kaameranurga ja pakib vide kokku ülekande efektiivusega arvestades. Peamiselt on vaatluse all reaalajas IMVV vood, ja vaatevahetust koheldakse diskreetse Markovi ahelana, kus üleminekütöenäosused arvutatakse Zipf jaotuse abil. Vältimaks edasi-tagasi aja (RTT) ülekandeviivitust, määratakse kohanduvad kvantimise parameetrid (QP) allesjäänud liigsete kaadrite jaoks. Sedasi säilitatakse minimaalne viivitus, läbi videkvaliteedi vähendamise RTT jooksul. Testimise tulemused näitavad, et antud meetod on parem, nii PSNR-i kui ka vahetusviivituse poolest, kui teised meetodid.

CERCS codes: Pilditehnika (T111)

Võtmesõnad: Interaktiivses mitmevaatelises videos, Zipf jaotuse, Hidden Markov Model, Vältimaks edasi-tagasi aja.

Contents

List of Figures

List of Tables

Abbreviations

IMVV - Interactive MultiView Video

RTT - Round Trip Time

MVV - MultiView Video

HEVC - High Efficiency Video coding

MV-HEVC - MultiView High Efficiency Video coding

3D-HEVC - Three Dimensional high Efficiency Video coding

HD - High Definition

3D - Three Dimensional

MV - MultiView

MVD - Multiview + Depth

AVC - Advanced Video Coding

TV - TeleVision

S3D - Stereoscopic Three Dimensional

HVS - Human Vision System

GOP - Group Of Picture

POC - Picture Order Count

CTU - Coding Tree Unit

CU - Coding Units

TU - Transform Unit

CB - Coding Block

PB - Prediction Block

DSC - Distributed Source Coding

MVC - MultiView Coding

SVC - Scalable Video Coding

MCS - Modulation and Coding Scheme

VMAG - View and Modulation coding scheme AGgregation

DIBR - Depth Input Based Rendering

RoI - Region of Interest

HMM - Hidden Markov Model

QP - Quantisation Parameter

PSNR - Pick Signal to Noise Ratio

CDF - Cumulative Density Function

LTE - Long Term Evolution

GUI - Graphical User Interface

P2P - Peer to Peer

MCS - Minimal Consistent Set

VMS - View and MCS Selection

VMAG - View and MCS Aggregation

FVTV - Free Viewpoint TV

1 Introduction

1.1 Problem overview

IMVV is constructed with the views captured by multiple cameras from different point of views and represents them using the display devices. So as compared to real life perception, users can change their point of view rather watching the entire video from a fixed angle.

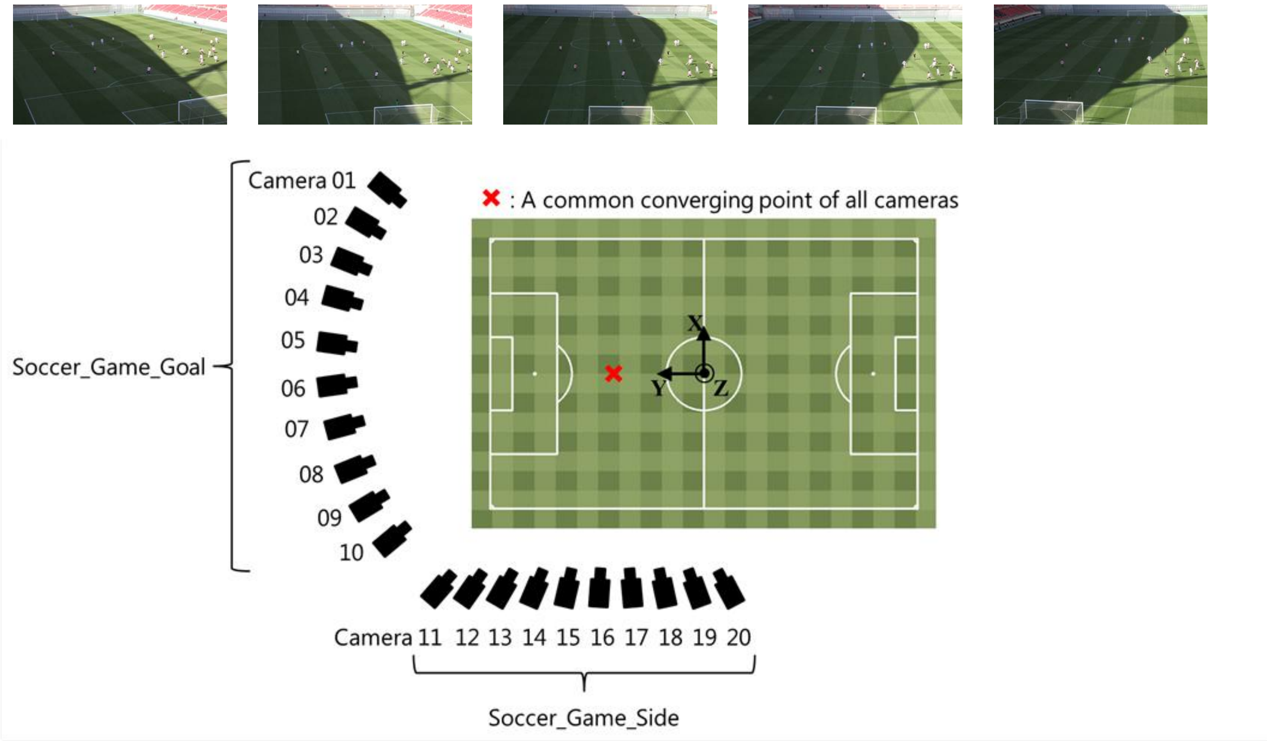
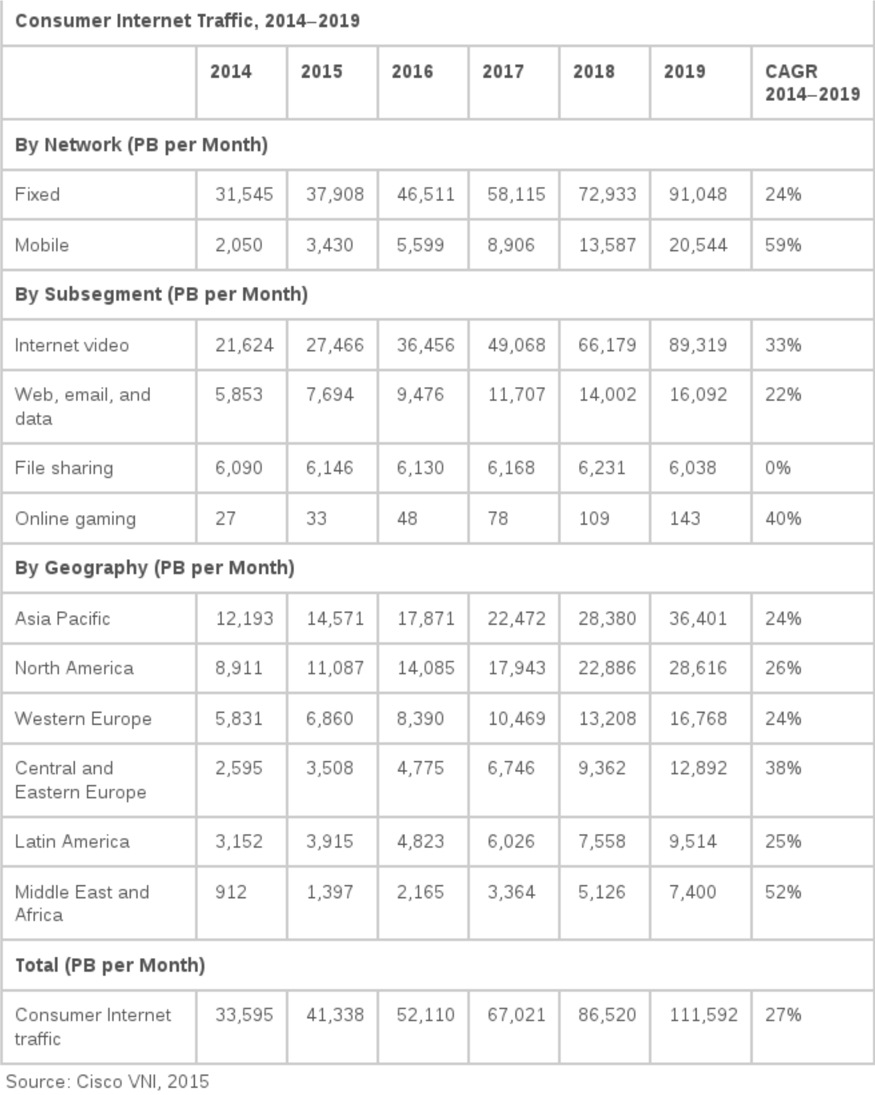

Camera 01
02
03
04
05
06
07
08
09
10
Soccer_Game_Goal
✖ : A common converging point of all cameras
X
Y
Z
Camera 11 12 13 14 15 16 17 18 19 20
Soccer_Game_Side

Figure 1.1: An example of 5 views of soccer game, source: [1] [2]

Generally when user switches the viewpoint, the information of view switching request need to transmit to the server side, which then start transmission of the desired view through the network channel to the user, that cause the delay in viewing, which is termed as RTT delay.

With the progress of technology, multimedia, video display device and network transmission methods, the need of comfort while switching views in MVV is one of the issue, considering current bandwidth availability.



Consumer Internet Traffic, 2014–2019							
	2014	**2015**	**2016**	**2017**	**2018**	**2019**	**CAGR 2014–2019**
By Network (PB per Month)							
Fixed	31,545	37,908	46,511	58,115	72,933	91,048	24%
Mobile	2,050	3,430	5,599	8,906	13,587	20,544	59%
By Subsegment (PB per Month)							
Internet video	21,624	27,466	36,456	49,068	66,179	89,319	33%
Web, email, and data	5,853	7,694	9,476	11,707	14,002	16,092	22%
File sharing	6,090	6,146	6,130	6,168	6,231	6,038	0%
Online gaming	27	33	48	78	109	143	40%
By Geography (PB per Month)							
Asia Pacific	12,193	14,571	17,871	22,472	28,380	36,401	24%
North America	8,911	11,087	14,085	17,943	22,886	28,616	26%
Western Europe	5,831	6,860	8,390	10,469	13,208	16,768	24%
Central and Eastern Europe	2,595	3,508	4,775	6,746	9,362	12,892	38%
Latin America	3,152	3,915	4,823	6,026	7,558	9,514	25%
Middle East and Africa	912	1,397	2,165	3,364	5,126	7,400	52%
Total (PB per Month)							
Consumer Internet traffic	33,595	41,338	52,110	67,021	86,520	111,592	27%

Source: Cisco VNI, 2015

Figure 1.2: Consumer Internet Traffic, 2014-2019, source : [3]

Video coding is also one of the most important challenge in order to transmit data efficiently through utilizing bandwidth properly. As per the Internet traffic data analyze done by Cisco as in Figure 1.2 [3], currently, Internet video is consuming 69.95% of bandwidth and by 2019 it will be consuming 80.04% of bandwidth. Therefore, efficient video coding techniques will light the burden of bandwidth and provide efficient media services or more consumer using ongoing networking technology without network problems (*e.g.,* network congestion). In other word the same quality of video if coded in much less size, it can reduce total bandwidth. Today, we are more concerned on video from single camera view. However, in case of multiview, when more than one camera are involved, data transmission will not be possible with the technological bottleneck of the current available data bandwidth.

In addition to the increment of video dimension, enlarged image resolution is a challenging problem for video transmission due to the high bandwidth requirement. For instance, most of the current movies have been started delivering as an 8K resolution. Figure 1.3 shows a comparative video size in recording and storing [4, 5].

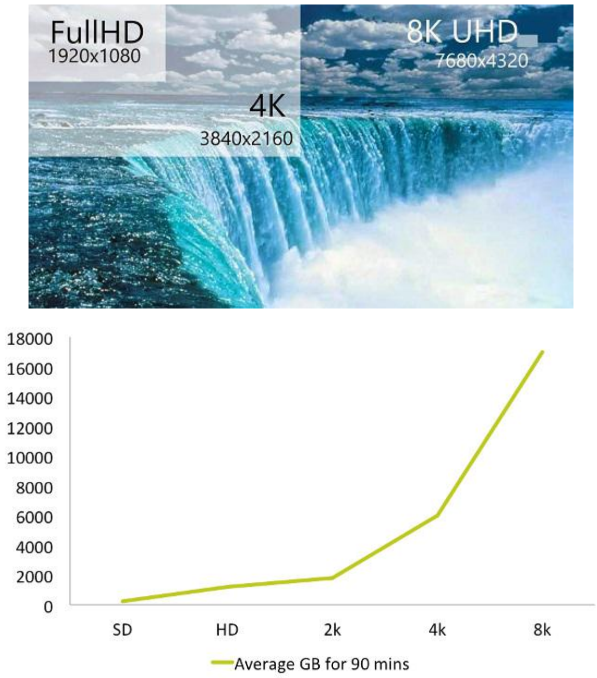


Figure 1.3: HD-8K comparison in recording screen size (up) and storing size (down), source : [4, 5]

1.2 Goals

These challenges in IMVV streaming and video coding can be categorised in broadly two parts [11], [12], that has been targeted here to overcome.

- Reducing the overall transmission bit-rate.
- Decreasing the transmission latency due to user interaction of choosing different viewpoints.

In this research of IMVV, the primary focus is to provide smooth view selection, when user does view-switch among available views. Due to the limitation of network bandwidth view-switch prediction is a required. This method predicts the possible view to which user may switch and also transmit them with the current view, so that it will not cause RTT delay and immediate switching among available views are possible at the receiver-end.

Bitrate optimisation of MVV compression is also necessary to achieve transmission/coding efficiency. Compression of views are done by maintaining current coding standard of MV-HEVC and 3D-HEVC.

In brief, by using more dynamic way to allocate the views of MVV to the user side, and using the state-of-the-art coding technique, current method provides user a more comfortable superior experience and let them involve with the context of video is our goal.

In first part of the thesis, some brief overview of the background has been provided in context of IMVV and Video Representation. Followed by related study, where the discussion about other related research work has been done. Next the methodology formulated in current research has been provided with algorithm of method and general flow diagram. Which is followed by the experimental results and finally conclusion including future work.

2 Background

In this method, while acknowledging the users' view switch request, predicted view is already been transmitted through network with the current view. That minimise delay in view switch. The transmission bit rate is also dynamically allocated with the standard video coding. In very simple way it can be depicted as Figure 2.1

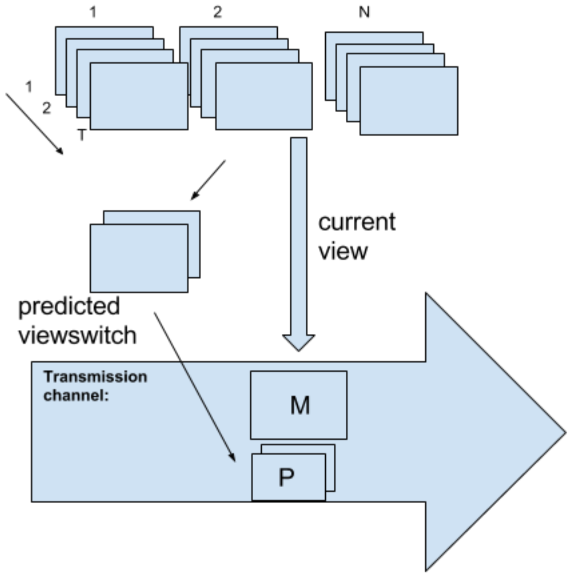


Figure 2.1: Overview of total process [M-current view, P - predicted view, N - total views.]

Recent advances in multimedia have paved the way for the recent MVV communication technologies. Most common examples include IMVV streaming [13,14], where the client can interact with the captured content by selecting the viewpoints at any position, and non-interactive MVV streaming, where the clients do not interact with the received content from the server, as in TV broadcast.

MVV consists several camera views taken from different perspectives. In the acquisition of MVV content, camera setup can be a 1D parallel array of N cameras [15], or 2D array of cameras positioned in M×N array [16], where M and N are the number of cameras in row and column directions, respectively. Also, it is possible to use cameras in a circular pattern, so a 360^o video experience can be achieved by stitching the captured MVV [17].

Contemporary video technology have made it feasible to stream 3D video at homes. The 3D video has had a significant influence on the movie industry, and public interest in Stereoscopic 3D (S3D) content has increased over the past decade. In the S3D technology, two slightly different views are presented to the eyes, and clients are able to perceive the 3D effect as those images are merged by the Human Visual System (HVS). Various technologies are used to facilitate 3D viewing, such as polarisation-based, shutter, and anaglyph technology.

The 3D experience is further enhanced using MVV, which in general includes more than two views, on multi-view displays (*e.g.,* auto-stereoscopic displays), where motion parallax and glasses-free user experience are realized.

Figure 2.2 shows the flowchart of an end-to-end S3D/MV video video streaming scenario. In order to achieve high performance of S3D/MV video streaming, the system needs to take all aspects into account. Hence, 3D scene representation, coding, transmission, video rendering and display technologies need to be efficiently optimized together. Depending on the application scenario, various methods are utilized in each of the building blocks.

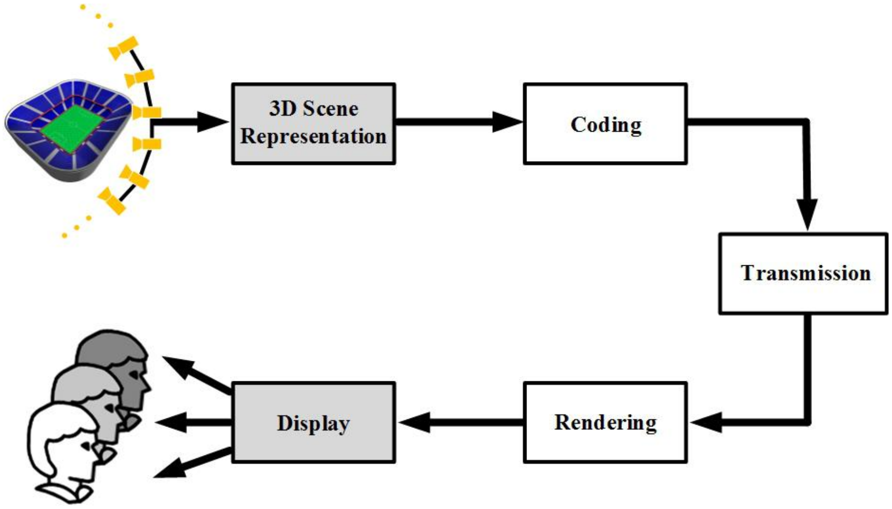
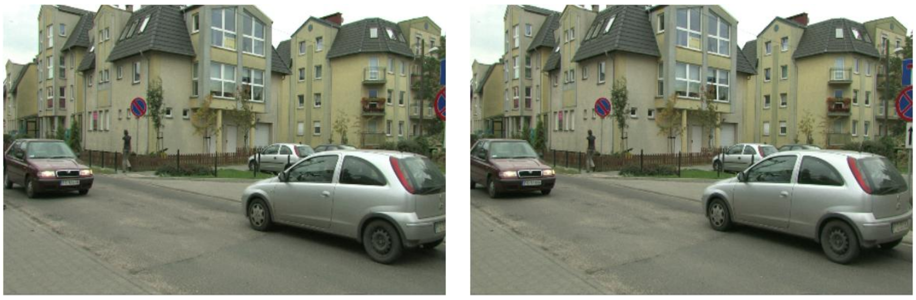


Figure 2.2: Stereoscopic 3D and multi-view video streaming chain from its capture to the end-user display, source : [6]

2.1 Video Representations

2.1.1 Stereoscopic 3D



(a) Left view (b) Right view

Figure 2.3: S3D video representation: example of the left and right views from the *PoznanStreet* test sequences.

The S3D video is the simplest, most cost efficient, and most widely used representation of 3D video. S3D consists of the left and right view pairs of the same scene as illustrated in Figure 2.3. These two views are captured with slightly different viewing angles due to the separation of eye.

2.1.2 Multi-View Video

MVV is another 3D representation that is required to support various 3D applications and displays. As illustrated in Figure 2.4, the MVV consist of more than two views of the same scene. Several views are captured simultaneously with calibrated cameras and provided to the user with the help of 3D displays.

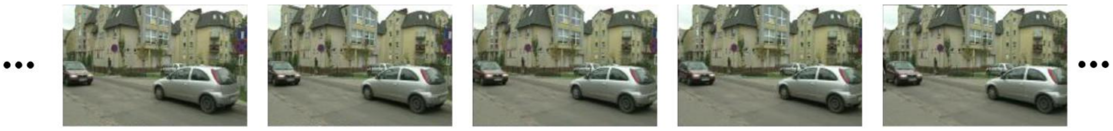


Figure 2.4: MVV representation: an example of *M* views from the *PoznanStreet* test sequence.

MVV was widely recognized as a powerful video format, and although 3D coding techniques offered some promising coding efficiency results, the high number of view required to provide significant end-user satisfaction, is accounting for extremely high transmission bit rates. This compromise of required transmission bit rate versus the perceptual quality level at the user-end lead to the deployment of the Multi-View plus Depth (MVD) format.

2.1.3 Multi-View plus Depth

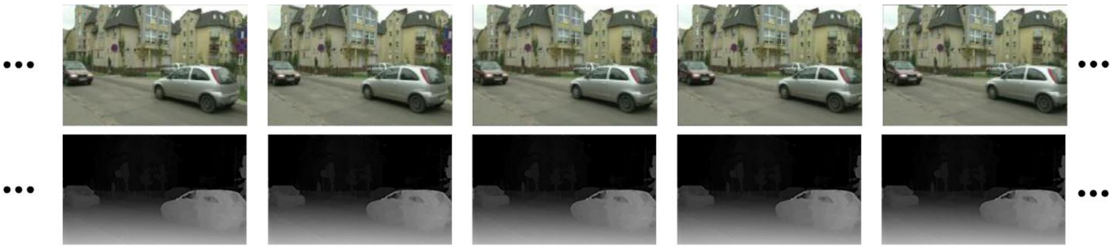


Figure 2.5: Multi-view Video plus-Depth (MVD) representation: an example of *M* views and corresponding depth maps from the *PoznanStreet* test sequence.

As illustrated in Figure 2.5, MVD contains multiple color views with the associated depth maps. The main advantage of using the MVD representation is that virtual (non-existing) views can be synthesised from the available reference views. Hence, generation of virtual views enables the users to change the viewpoint within the scene freely. On the one hand, this ability can enhance the immersive user experience.

2.1.4 3D Video Coding

Figure 2.6 [6] shows the life-cycle of video codec, and by experimenting various parameters involved in compression efficient coding can be done, which again provide better transmission over network or storage.

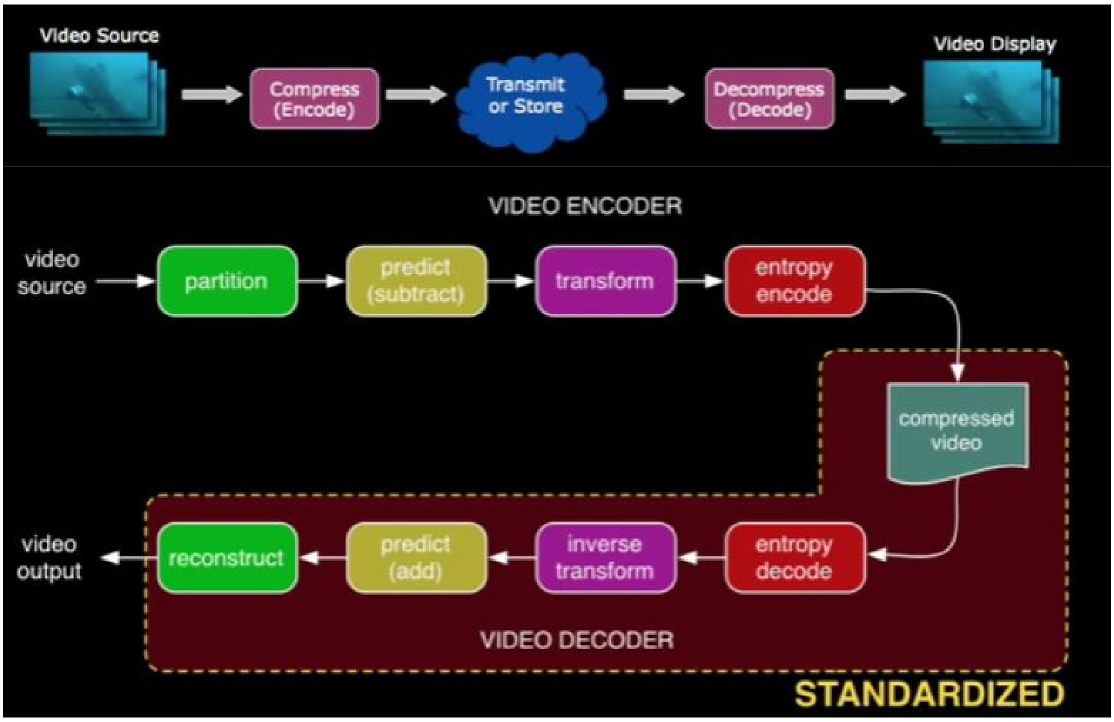


Figure 2.6: Video coding lifecycle, source : [6]

H.264/AVC video coding standard was introduced in 2003, and 10 years after the AVC standard, new coding standard H.265/HEVC came in action. Comparative analysis on those coding techniques has been conducted, and mentioned later part of the thesis.

In order to understand the coding parameters, though there are lots of them and each of them has significant impact on coded result, mainly parameters involved in partition and prediction steps are explained below sections.

Partition:

In video coding, video input is first divided into image frames with the correct frame rate as per the video has been captured. Figure 2.7 [7] provide reference of how I, P and B slices are used in coding. where Group of Picture abbreviated as GOP and Picture Order Count abbreviated as POC.

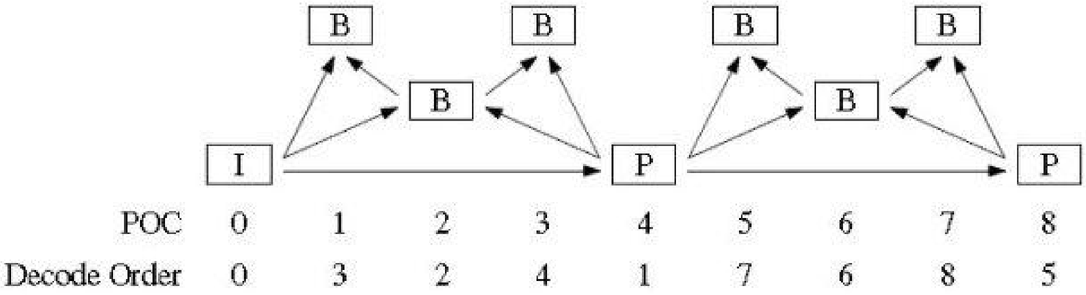


Figure 2.7: HEVC encoder partition , source : [7]

Each image frame has been partitioned into slices, and slices are the sequence of Coding Tree Units (CTU). Slices have an important role while video transmission occurs. They hold the information of synchronization, which takes effect in case of packet loss. Again slices can be categorized in three main types [6, 8]:

1. **I slice :** Where the Coding Units (CU) are coded using the *intrapicture* prediction.

2. **P slice :** Where the CUs are coded using the *interpicture* prediction with *uniprediction*

3. **B slice :** Where CU are coded using the *interpicture* prediction with *biprediction*

Figure 2.8 shows the CTU, CU and partition example over a image frame.

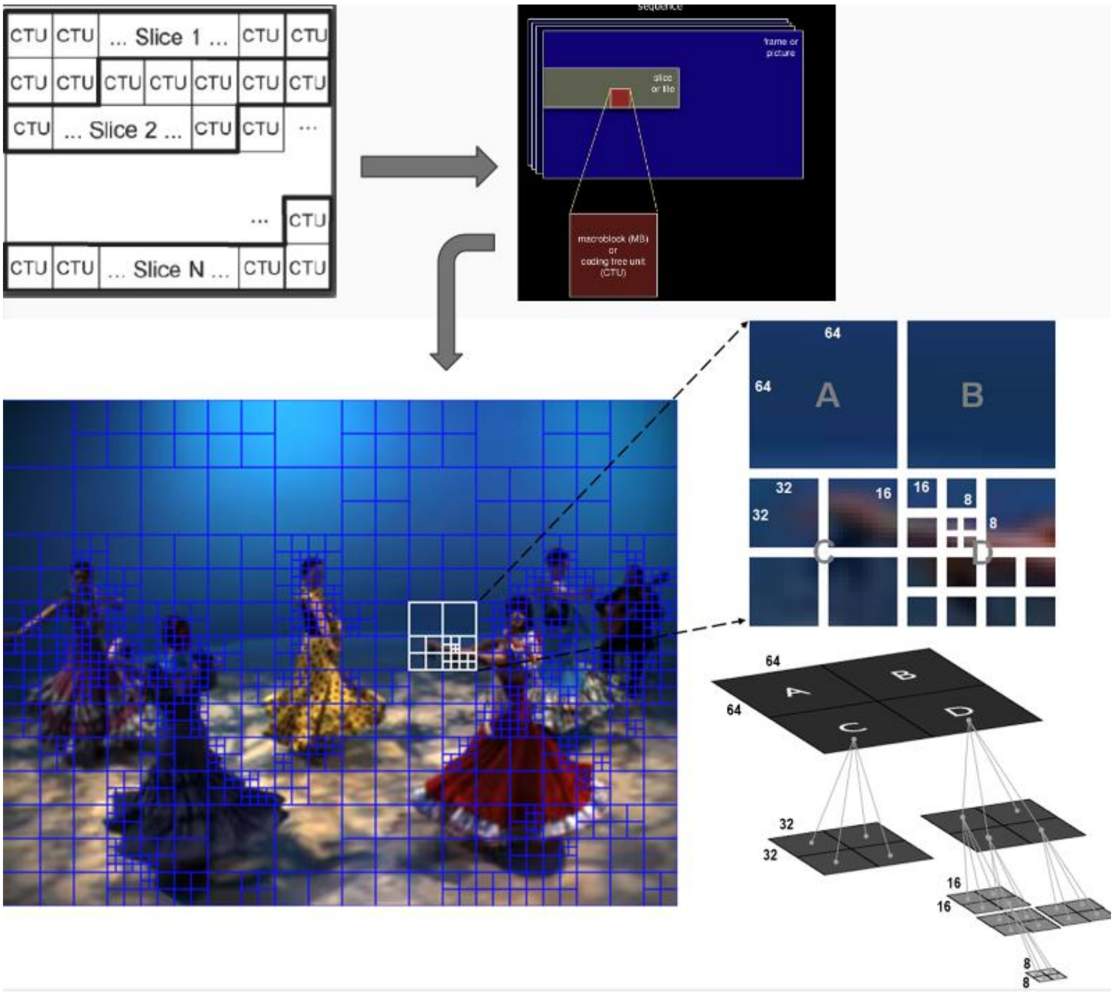

CTU CTU ... Slice 1 ... CTU CTU
CTU CTU CTU CTU CTU CTU CTU
CTU ... Slice 2 ... CTU CTU ...
... CTU
CTU CTU ... Slice N ... CTU CTU
sequence
frame or picture
slice or tile
macroblock (MB) or coding tree unit (CTU)
64
64
A
B
32
32
16
16
8
8
C
D
64
64
A
B
C
D
32
32
16
16
8
8

Figure 2.8: HEVC encoder partition, source : [6, 8]

Prediction:

Each CU is split into one of 8 partition modes $(2Nx2N, 2NxN, Nx2N, NxN, 2NxnU, 2NxnD,\ nLx2N, nRx2N)$. Which can be shown by Figure 2.9 [8].

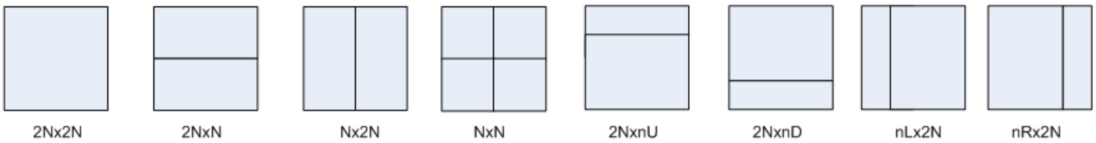


Figure 2.9: Mode of splitting CB to PB

Prediction can be categorized in two types.

- **Intra Prediction** - When a CU follows exactly the TU tree.
- **Inter Prediction** - Motion vector prediction, Motion compensation

Figure 2.10 [6] shows how in CU prediction is done considering directions.

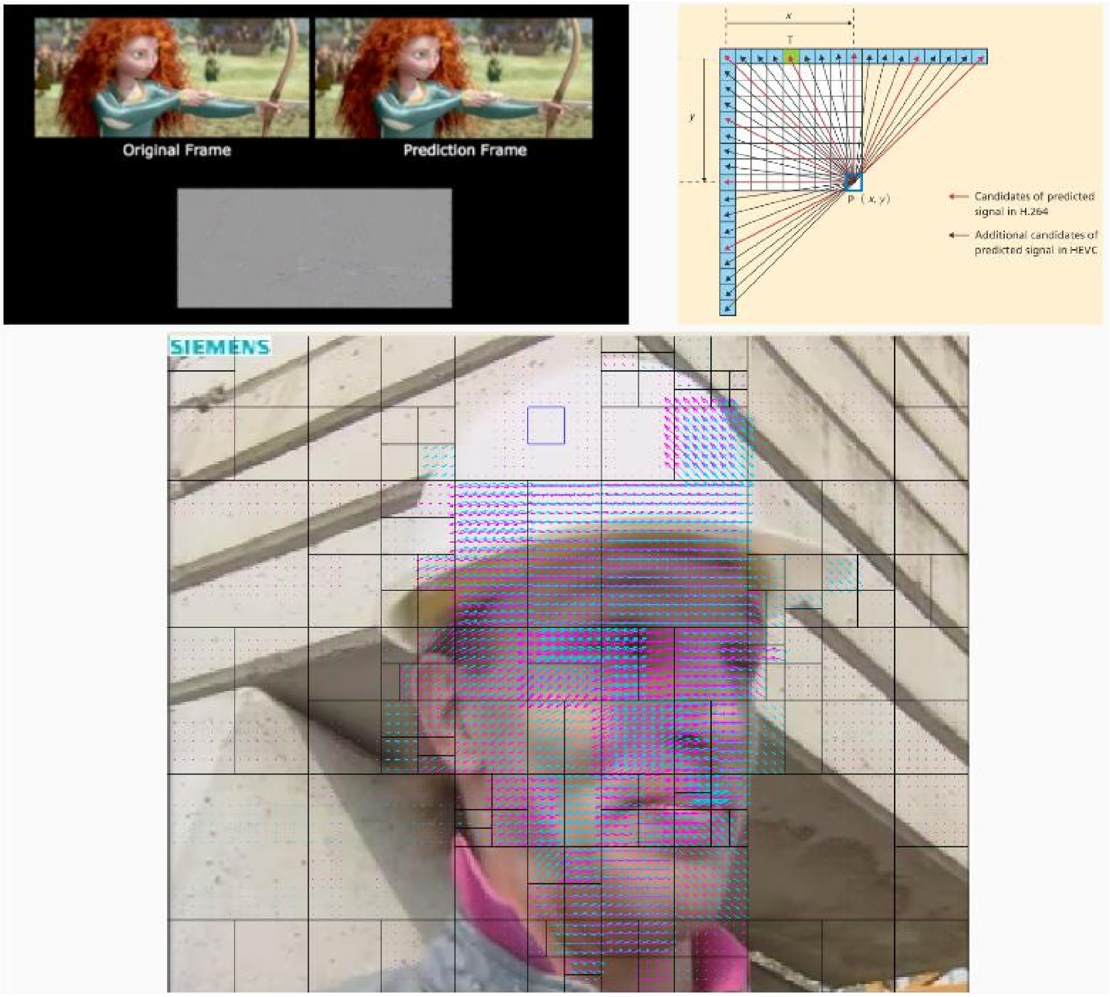

Original Frame
Prediction Frame
x
T
y
P (x, y)
Candidates of predicted signal in H.264
Additional candidates of predicted signal in HEVC
SIEMENS

Figure 2.10: HEVC encoder prediction, source : [6]

The first version of HEVC standard finalized in April 2013, The reference software is called HM-HEVC (HEVC Test Model), the second version and the third versions include MV-HEVC and 3D-HEVC, respectively.

HM-HEVC

HM encoder and decoder are just common reference implementation of an HEVC encoder and decoder, for testing and evaluating the technology for independent encoding and decoding.

MVC and MV-HEVC

An extension of Advance Video Coding (AVC) standard for MVV and Multi-View extension of High Efficiency Video Coding (MV-HEVC) were released to support the simultaneous compression of video from several cameras. Since all captured views in MVV content represent the same scene from varying perspectives, they contain a considerable amount of inter-view statistical redundancies. The straightforward solution for the S3D/MVV would be to encode all views independently using a conventional video coding such as HEVC. To this end, MVC and MV-HEVC standard, which uses a hierarchical B-frame structure with the addition of inter-view prediction modes, encoded number of views and produces a cost-efficient bit-stream.

3D-HEVC

As mention before, 3D video is mostly represented by MVD format, in which a group of captured views and associated depth maps can be transmitted efficiently. The 3D-HEVC standard exploits correlation between the MVD sequences in a manner to the MVC/MV-HEVC standard. It also provides scalable coding in a way so that the subset of the views can be extracted by discarding NAL units from the bit-stream. In this way, a subset of the views can be independently decoded with this coding standard.

Experimental results demonstrate that 3D-HEVC achieves the highest 3D video coding efficiency relative to the state-of-the-art video coding standard, i.e. HEVC and MV-HEVC. Experimental results in the literature found that 3D-HEVC achieves more than 40% bitrate saving compared to HEVC.

3 Related Study

In the literature there exists some research work focusing on achieving video coding efficiency. Redundant P-frame and Distributed Source Coding (DSC) frames have been used for coding structure [18]. Heuristic based distributed and cooperative replication strategy are adopted to take advantage of the correlations between the multiple views for source effective content delivery [19].

Kalman-filter based head position prediction has been used to minimise view switch delay [20]. Also, network bandwidth has been saved by adopting Multi-View Coding (MVC) and Scalable Video Coding (SVC) concept. In [21], the method uses attention-weighted bit-rate allocation technique, which again dependent on the number of users engaged at a certain time.

Later, view and Modulation and Coding Scheme (MCS) aggregation (VMAG) are proposed in [22] to find the optimal solution of view and MCS selection problem, which deals on synthesising view based on Depth-Image-Based Rendering (DIBR) [23, 24].

Overview of HEVC video coding with technical parameters has been described very nicely in [8], wchi provides much knowledge in the field of video coding. The concept of video coding is described and comparative analyzed in [25] chronologically from AVC to recent HEVC. Also the complexity of coding techniques and implementation was clearly explored in [26].

Frame structure optimisation has been performed in [27, 28] in order to reduce transmission rate using a low-complexity greedy algorithm. According to their experiment proposed method gives around 42% lower transmission rate than I-frame only structure. But the paper do not talk about time complexity and as their method need many iteration to optimise the Lagrangian cost, it may not be useful for real time media streaming. The challenge of prediction of desired view after viewswitch has been addressed in [29], and to overcome it user tracking and compression jointly has been proposed. Which has a dependency of additional hardware and can function unexpectedly in the presence of multiple people.

In [30] P2P streaming framework is proposed in multiview video that organizes viewers of different views together to cooperate in view switching and content delivery, achieving reduced view switching delay. But like other P2P system it has limitation when there are no neighbours, or all neighbours are having different views. And by cross view resource user mostly compromise with the desired viewpoint rather watches different crossview video based on neighbours.

In [31] different qualities (or bit-rate) are broadcasted based on number of viewers. The optimality of the method relies on knowing the viewpoint probability distribution at every time instance. View and MCS Selection Problem (VMS) used to minimize the bandwidth consumption for multi-view 3D video multicast in LTE networks, a new optimization problem is formulated in [32]. And an algorithm, called View and MCS Aggregation (VMAG) is proposed to find the optimal solution of VMS.

Another approach has been seen in [33] which extend the optimized bit allocation scheme to allow more generic camera arrangement in FVTV. Where again the quality(or bit-rate) at each camera was determined by viewer attention in order to minimize total observed distortion. In [34] a coding structure used based on redundant P-frames and distributed source coding (DSC) frames to achieve efficiency in coding, view switches and content replication. Also to take advantage of the correlation between the multiple views for resource-effective content delivery a heuristic-based distributed and cooperative replication strategy used.

In [35] bandwidth requirements is reduced by transmitting a small number of views selected according to users head position.It makes use of multiview coding (MVC) and scalable video coding (SVC) concepts together to obtain improved compression efficiency while providing flexibility in bandwidth allocation to the selected views.Another approach is seen in [36] by using redundant frame structure offering a range of tradeoff points between transmission and storage.

4 Motivation

The principle obstacles of IMVV has already been indicated in the introduction. In brief, reduction in transmission bit-rate and transmission latency initiated by viewpoint switch is the purpose of the proposed method. These are undesirable because of its time consumption. To the best of our knowledge, very limited research has been done to predict view switch and mostly they ignored the motive of users behind the switch.

In this paper, the proposed method is primarily inspired by the fact that the user is not switching between the views randomly and the user switches between views to find the best perception of objects in the video. Hence, in the proposed method the context (i.e., face) and phenomena (i.e., Region of Interests (ROI)) in each view are considered and accordingly a novel method to predict view switch based on them is introduced. Also, an efficient bit-rate allocation used with the state-of-the-art compression technique to minimise the transmission bit-rate. In case user switched to an extreme view, which were not predicted, as it is not based on the concept of users likely-hood to switch to some other view for better perspective of video. Still in order to continue with real time experience redundant frames with lower quality are transmitted along-with, So user can have always view available though with inferior quality, and quality is restored in RTT time-span.

5 Methodology

5.1 MultiView Video

This work is contributing by addressing two significant challenges. The first challenge is the selection of the view, and the second one is to predict the next view. To overcome the first challenge, in this research work, the view that has most action and contains more ROI, such as the face, is selected. It is important to note that we assume that the user is most interested in viewing from the camera which provides maximum information about the actions and ROIs among several camera views.

In order to overcome the second part of the challenge, in this paper, the prediction is done on the possible viewpoint that user may choose. In order to switch between the views smoothly, intermediate camera views between the current view and the predicted view are also transmitted. Also, we transmit the compressed predicted views and also redundant views with adaptive compression.

The proposed method is divided into five main steps which are listed below.

1. Selection of initial camera.
2. Selection of proceeding cameras.
3. Prediction of user viewpoint switch.
4. Compression of viewpoint switch camera views.
5. Adaptive compression of redundant camera views.

The aforementioned steps are described in details in the following parts.

5.1.1 Selection of initial camera

To select the first view out of N camera views, there are two possible ways. First, the initial view can be selected by user interaction where the user provides the starting camera view. Second, the initial view can be selected by calculating the camera view that consist of the greatest salient features and contexts. This information can be calculated by using equation 5.1.

$$F_i = a_i + f_i \tag{5.1}$$

where F_i is the number of features in i^{th} camera view, a_i denotes the number of action, and f_i represents the number of RoI, which in this work is the number of detected faces. The number of action is decided by considering the salience [37] difference between two consecutive frames of same camera view, as can be expressed by equation 5.2.

$$a_i = s_{c(i)} - s_{c(i-1)} \tag{5.2}$$

where $s_{c(i)}$ is the salience feature of i^{th} frame of camera view c. Hence, camera view consisting most features has been determined as follows:

$$F_{max} = max(F_i) \quad , \quad i \in \{1, \ldots, N\} \tag{5.3}$$

The camera view with maximum features is nominated as the initial camera view to begin with.

5.1.2 Selection of proceeding cameras

The selection process of proceeding camera views are done similar to the process of selection of the initial camera view. The main change between these two steps is on step size. In both parts the action is determined by the salience component [37] moved in two consecutive frames of the luminance component of the captured camera view through equation 5.1, 5.2 and 5.3. The context (i.e., face) is determined by Viola-Jones face detection method [38]. Unlike preceding part while selecting the next camera angle, the step size is defined as user input 's' and the camera is chosen which lies inside the user specified step size.

If the current chosen camera is 'n' then equation 5.1 to 5.3 can be modified as follows.

$$F_{max} = max_s(a_i + f_i) \tag{5.4}$$

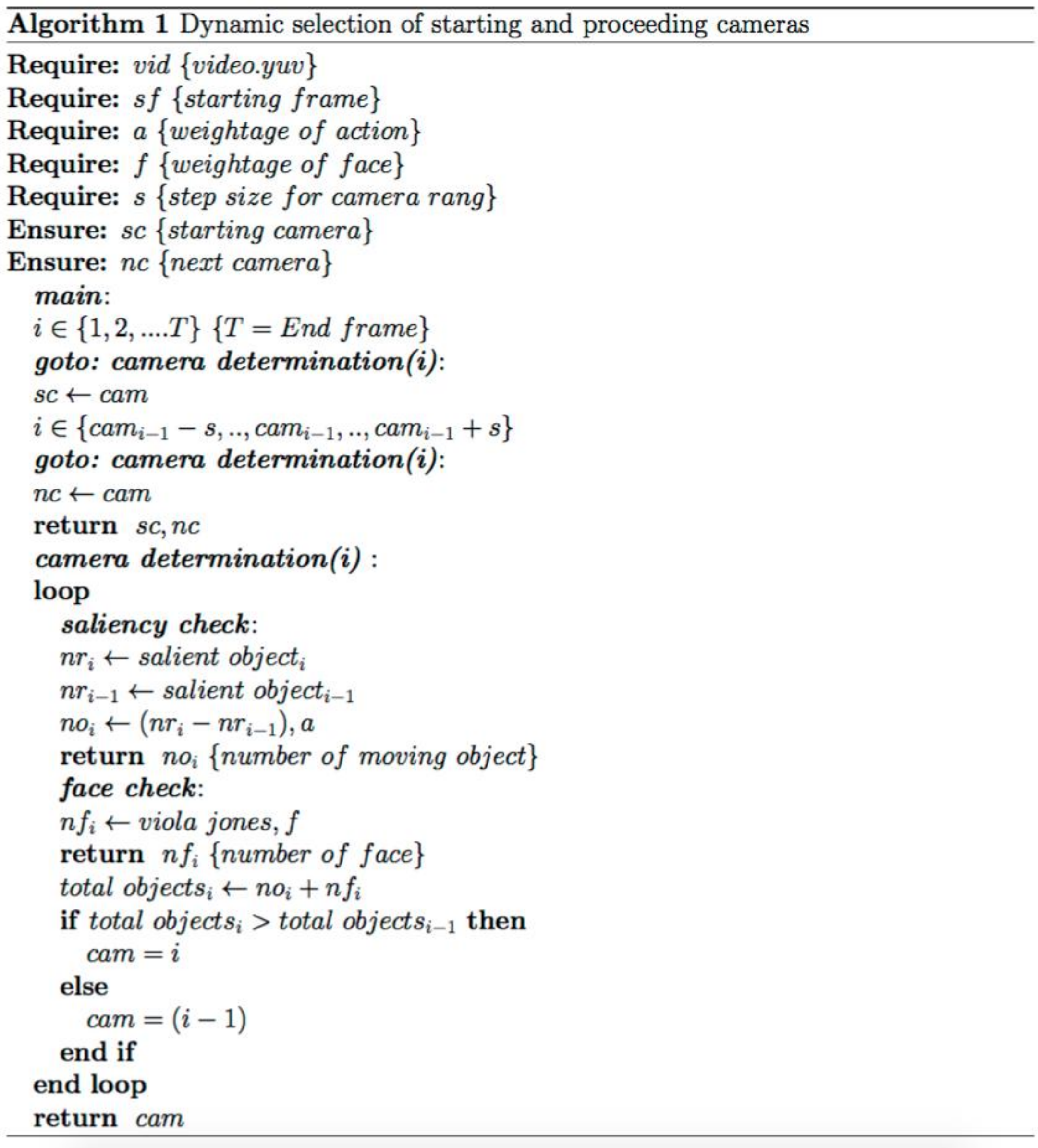
where s is a set of cameras as denoted below.

$$s \in \{(n-s), ...s, ...(n+s)\} \tag{5.5}$$

In the proposed method in order to discard the discomfort, which may arise to the user due to frequent shifts in camera views and increase the quality of the output the next camera should be adopted only after 0.5 sec of the last camera view adjust. If there are t frames per second, the following equation is used to find the time when the next camera view should be inspected.

$$n_{i+1} = n_i + t/2 \tag{5.6}$$

To elaborate saliency check of camera views, [37] is adopted in the following form. In each frame, dyadic Gaussian pyramid subsample is performed to keep the original size, and then surround centre check is applied followed by normalisation on each frame. This process produces a saliency image from each frame. After the salience image is generated from both consecutive frames, for all saliency objects that changed its position is marked as moving object, which indicated as the actions in a frame of a view. The detail steps of the aforementioned two steps of the camera view selection is provided in algorithm 1.



Algorithm 1 Dynamic selection of starting and proceeding cameras

```
Require: $vid \{video.yuv\}$
Require: $sf \{starting\ frame\}$
Require: $a \{weightage\ of\ action\}$
Require: $f \{weightage\ of\ face\}$
Require: $s \{step\ size\ for\ camera\ rang\}$
Ensure: $sc \{starting\ camera\}$
Ensure: $nc \{next\ camera\}$
  main:
  $i \in \{1, 2, ....T\}$ $\{T = End\ frame\}$
  goto: camera determination(i):
  $sc \leftarrow cam$
  $i \in \{cam_{i-1} - s, .., cam_{i-1}, .., cam_{i-1} + s\}$
  goto: camera determination(i):
  $nc \leftarrow cam$
  return $sc, nc$
  camera determination(i) :
  loop
    saliency check:
    $nr_i \leftarrow salient\ object_i$
    $nr_{i-1} \leftarrow salient\ object_{i-1}$
    $no_i \leftarrow (nr_i - nr_{i-1}), a$
    return $no_i$ $\{number\ of\ moving\ object\}$
    face check:
    $nf_i \leftarrow viola\ jones, f$
    return $nf_i$ $\{number\ of\ face\}$
    $total\ objects_i \leftarrow no_i + nf_i$
    if $total\ objects_i > total\ objects_{i-1}$ then
      $cam = i$
    else
      $cam = (i - 1)$
    end if
  end loop
  return $cam$
```

5.1.3 Prediction of user viewpoint switch

While the current view is selected by the process as mentioned above and is transmitted, the user can change camera viewpoints at one's convenience. Then essentially a feedback signal should come to the server of changed viewpoint ensure server's processing of the new viewpoint to user, that will induce some delay in the user's selected view.

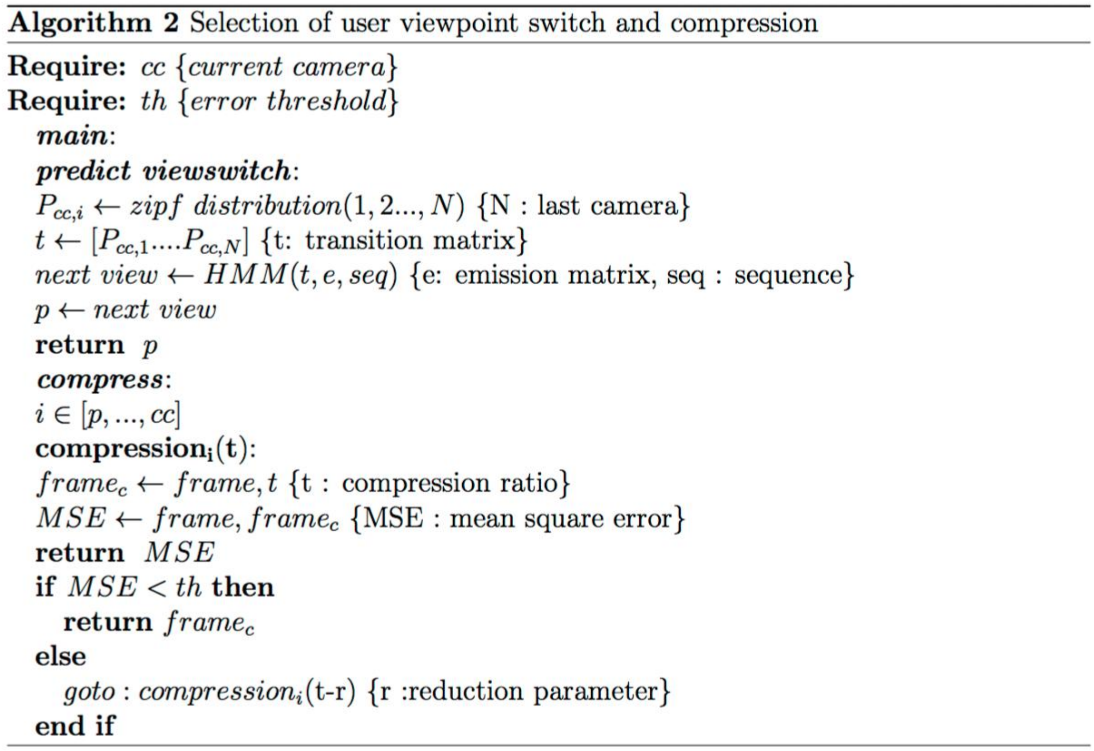
To minimise this delay, a novel way of anticipation of viewpoint switch is proposed in this work. Hidden Markov Model (HMM) is adopted as a plinth of the prediction strategy [39]. The hidden states are the all possible camera views indeed. The transition matrix formulated utilising the probabilities of change among states, and those probabilities can be represented as follows:

$$P_{n,n\pm i} = P_j \times d_{n,n\pm i}^{-1} \tag{5.7}$$

where, P_j is zipf distribution [40] and d is the distance from the predicted camera view and current camera view, and the current view is denoted by i.

5.1.4 Compression of camera view

The current camera view with the predicted view as calculated by the previous step is supplied to the transmission channel. Which in case of sudden view change to the already predicted views will minimise the view switch time by rendering the predicted view along with intermediate views, which is also supplied to the transmission medium after applying the High Efficiency Video Coding (HEVC) standard [41]. Method [24] is used for calculation of the bit-rate. Last two steps of the proposed method are replicated in algorithm 2.



Algorithm 2 Selection of user viewpoint switch and compression

```
Require: $cc$ {$current\ camera$}
Require: $th$ {$error\ threshold$}
  main:
  predict viewswitch:
  $P_{cc,i} \leftarrow zipf\ distribution(1, 2..., N)$ {N : last camera}
  $t \leftarrow [P_{cc,1} .... P_{cc,N}]$ {t: transition matrix}
  $next\ view \leftarrow HMM(t, e, seq)$ {e: emission matrix, seq : sequence}
  $p \leftarrow next\ view$
  return $p$
  compress:
  $i \in [p, ..., cc]$
  $\mathbf{compression_i(t)}$:
  $frame_c \leftarrow frame, t$ {t : compression ratio}
  $MSE \leftarrow frame, frame_c$ {MSE : mean square error}
  return $MSE$
  if $MSE < th$ then
    return $frame_c$
  else
    goto : $compression_i$(t-r) {r :reduction parameter}
  end if
```

5.1.5 Adaptive compression of redundant camera views

In order to satisfy viewswitch request when user has chosen some different view rather than predicted one. Proposed method also transmit redundant views with high lossy compression. The quality of switched view will be established after RTT. The QP of redundant views has been determined based on total available transmission bandwidth. Where bandwidth information can be achieved from network specification. The available bandwidth is utilised for transmitting redundant views of streaming video. By performing an experiment with several QP, and measuring bitrate Figure 5.1 plot has been produced.Next QP can be adjusted of the frames to attain desired bit stream size. It can be represent mathematically as below in equation 5.8:

$$q = \frac{1}{A}\Big[\frac{Bv}{100} - (t_1 + t_2)\Big] \tag{5.8}$$

where,
q is bitstream size(Kb) ,
A is number of redundant frames,
B total bandwidth as per network specification,
v is available of percentage of total network bandwidth for video streaming,
t_1 is the bandwidth taken for current active view,
t_2 is the bandwidth taken by predicted views as calculated in previous section
also all calculation done for each second, so how many frames are compressed will be depended on framerate of the video. All information about the database taken into consider for this paper experiments can be found in Section 6.

$$qp = f(q) \tag{5.9}$$

Equation 5.9 shows the QP value (qp), which can be calculated from fitting function $f(q)$ from Figure 5.1. Here the fitting function in the figure has been calculated by using Piecewise cubic interpolant [42].
Also Figure 5.2 provides an idea of resulting quality of encoded video, in terms of PSNR as a relation of QP. Which shows that QP and PSNR are inversely proportional.

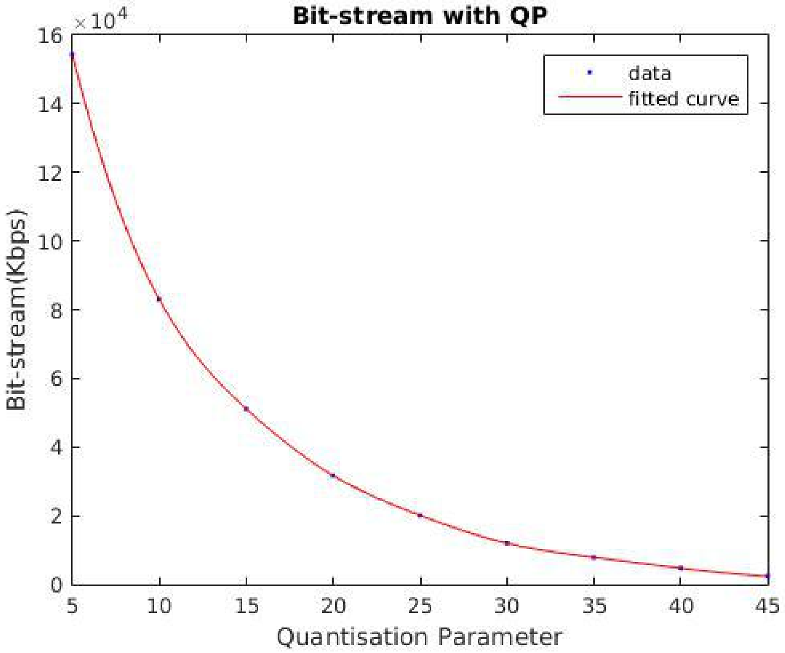


Figure 5.1: QP with bitstream

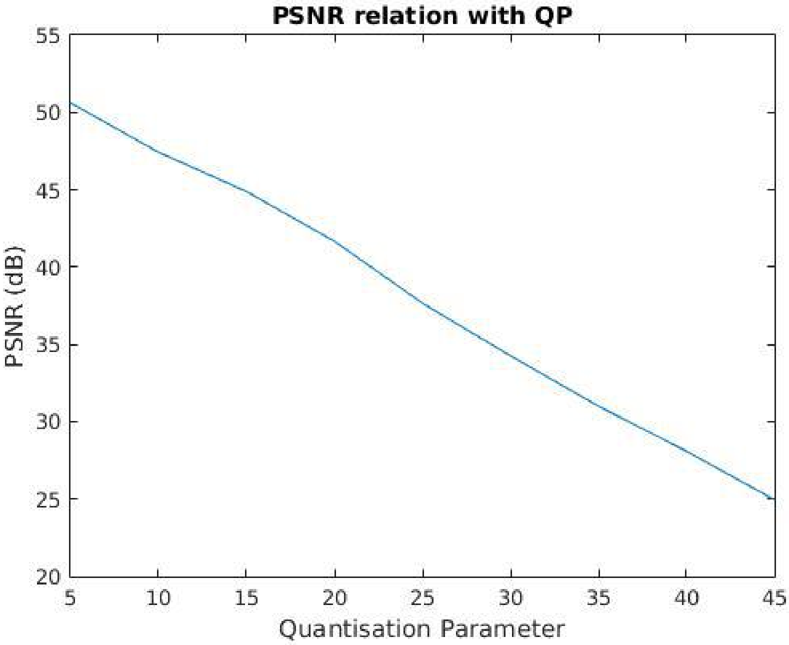


Figure 5.2: PSNR with QP

5.2 Flow Diagram

Figure 5.3 shows the flow of step 1, which is consist of first two parts as mentioned above named, Selection of initial camera and election of proceeding cameras. And Figure 5.4 shows the flow of step 2, which is consist of last two parts named, Prediction of user viewpoint switch and Compression of camera view respectively.

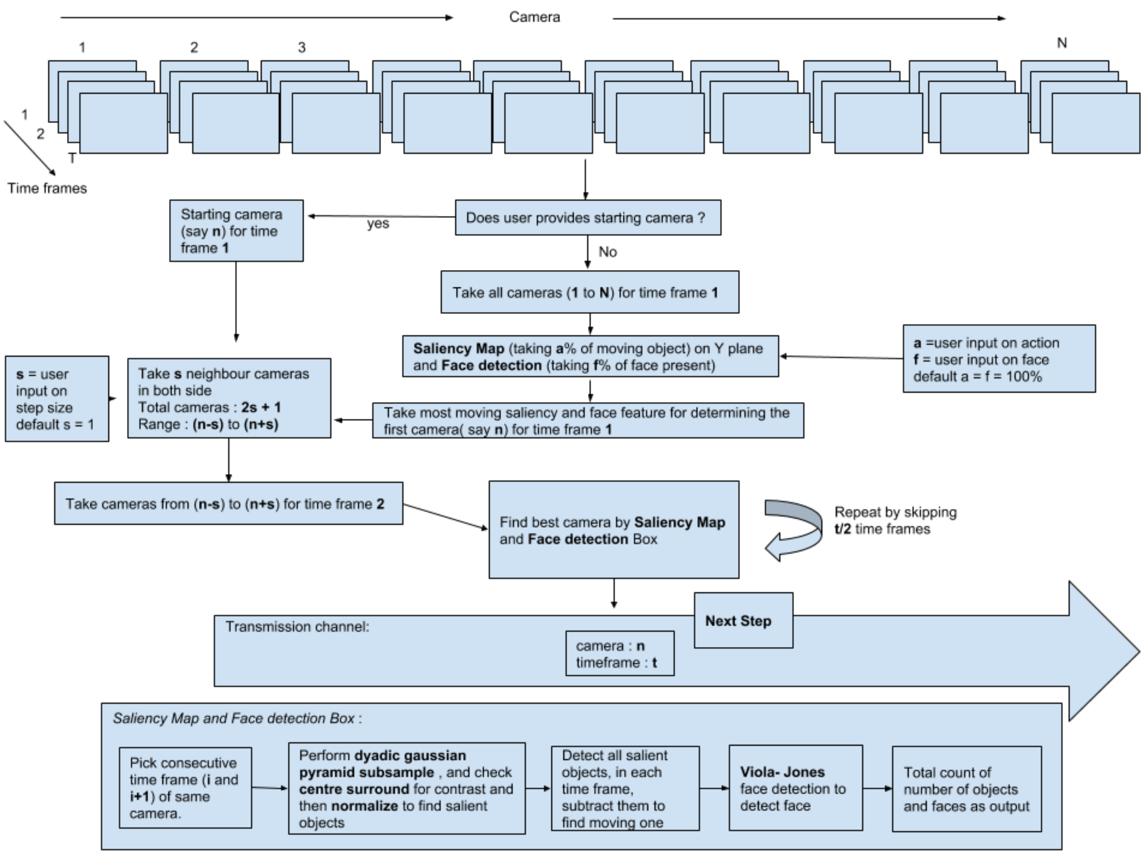

Camera
1
2
3
N
1
2
T
Time frames
Starting camera (say n) for time frame 1
yes
Does user provides starting camera ?
No
Take all cameras (1 to N) for time frame 1
Saliency Map (taking a% of moving object) on Y plane and Face detection (taking f% of face present)
a =user input on action
f = user input on face
default a = f = 100%
s = user input on step size default s = 1
Take s neighbour cameras in both side
Total cameras : 2s + 1
Range : (n-s) to (n+s)
Take most moving saliency and face feature for determining the first camera(say n) for time frame 1
Take cameras from (n-s) to (n+s) for time frame 2
Find best camera by Saliency Map and Face detection Box
Repeat by skipping t/2 time frames
Transmission channel:
camera : n
timeframe : t
Next Step
Saliency Map and Face detection Box :
Pick consecutive time frame (i and i+1) of same camera.
Perform dyadic gaussian pyramid subsample , and check centre surround for contrast and then normalize to find salient objects
Detect all salient objects, in each time frame, subtract them to find moving one
Viola- Jones face detection to detect face
Total count of number of objects and faces as output

Figure 5.3: Flow diagram of step 1

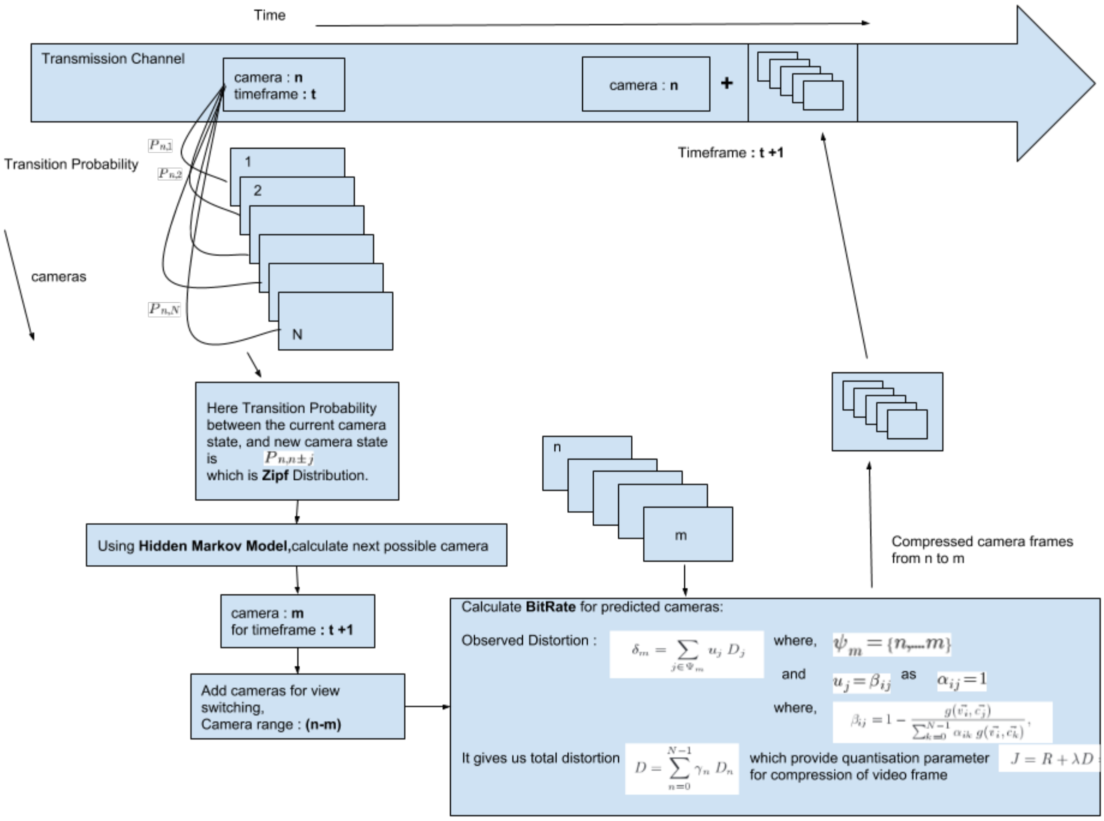

Time
Transmission Channel
camera : **n**
timeframe : **t**
camera : **n**
+
Timeframe : **t +1**
Transition Probability
$P_{n,1}$
$P_{n,2}$
$P_{n,N}$
1
2
N
cameras
Here Transition Probability between the current camera state, and new camera state is $P_{n,n\pm j}$ which is **Zipf** Distribution.
Using **Hidden Markov Model,**calculate next possible camera
camera : **m**
for timeframe **: t +1**
Add cameras for view switching,
Camera range : **(n-m)**
n
m
Compressed camera frames from n to m
Calculate **BitRate** for predicted cameras:
Observed Distortion : $\delta_m = \sum_{j \in \Psi_m} u_j D_j$ where, $\psi_m = \{n,...m\}$
and $u_j = \beta_{ij}$ as $\alpha_{ij} = 1$
where, $\beta_{ij} = 1 - \frac{g(\vec{v_i}, \vec{c_j})}{\sum_{k=0}^{N-1} \alpha_{ik}\, g(\vec{v_i}, \vec{c_k})},$
It gives us total distortion $D = \sum_{n=0}^{N-1} \gamma_n D_n$ which provide quantisation parameter for compression of video frame $J = R + \lambda D$

Figure 5.4: Flow diagram of step 2

6 Results

6.1 IMVV

Figure 6.1 shows few sample image frames from various camera angle from "Champagne tower" database available for testing for multiview video.

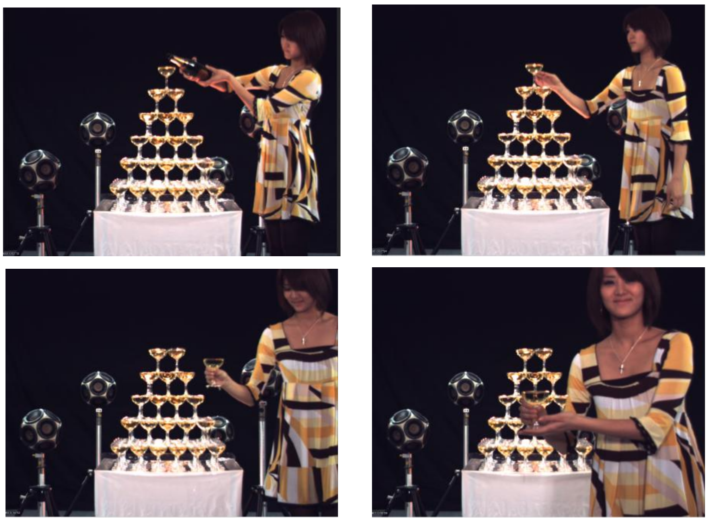


Figure 6.1: Champagne tower data-set from camera view 38 [9]

After applying the saliency map and face detection algorithm, as shown in 5.3 *Saliency Map and Face detection Box* saliency image with moving object and face in the frame can be labelled as ROI of the current frame. As an example of the output Figure 6.2 can be referred.

To evaluate the proposed method, following database [9] has been used, where cameras

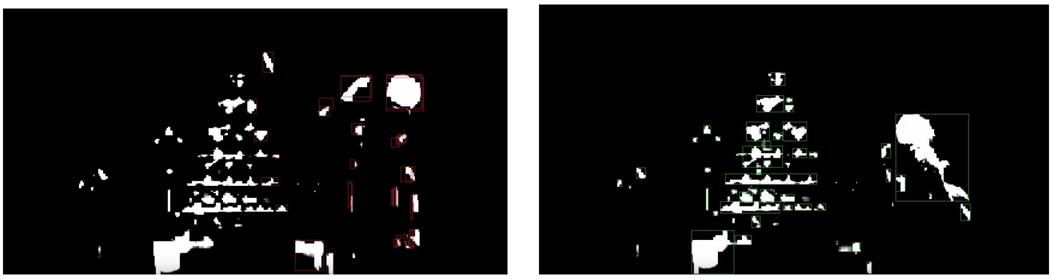


Figure 6.2: Saliency figure of moving objects from camera view 38

are positioned in an one dimensional array. A graphical user interface has been built for testing where user can provide own choice of initial camera view. If user does not want to mention any specific camera, it can be calculated automatically.

Experiments has been performed over three databases, out of which *champagne_tower* and *pantomime* have 500 frames and *dog* has 300 frames for each camera view. The camera array of the selected database are placed with a horizontal 50 *mm* interval and they converge at the centre of wall at 8.2 *m* from the array of cameras. The resolution of cameras is 1280×960 pixels with a frame rate of 29.4 *fps*.

Selection of starting camera has done as per the proposed method. When the user does not provide the camera view to start with, all of the camera views has been taken in consideration for the very first frame calculation.

The plot of Peak Signal to Noise Ratio (PSNR) over frames while viewswitch happens has been shown in Figure 6.3 and the CDF of viewswitching delay has been shown in Figure 6.4. The experiment result shows a good value of PSNR at range around 34 dB -38 dB for quantisation parameter (QP) value of 40 calculated as average of all camera frames. And the cumulative view switching delay mostly below around 1 *sec.*, this is because, the predicted frames are already in user side, so if user chose to switch view among those predicted frame, switching delay only counts for switching from current frame to new view frame.

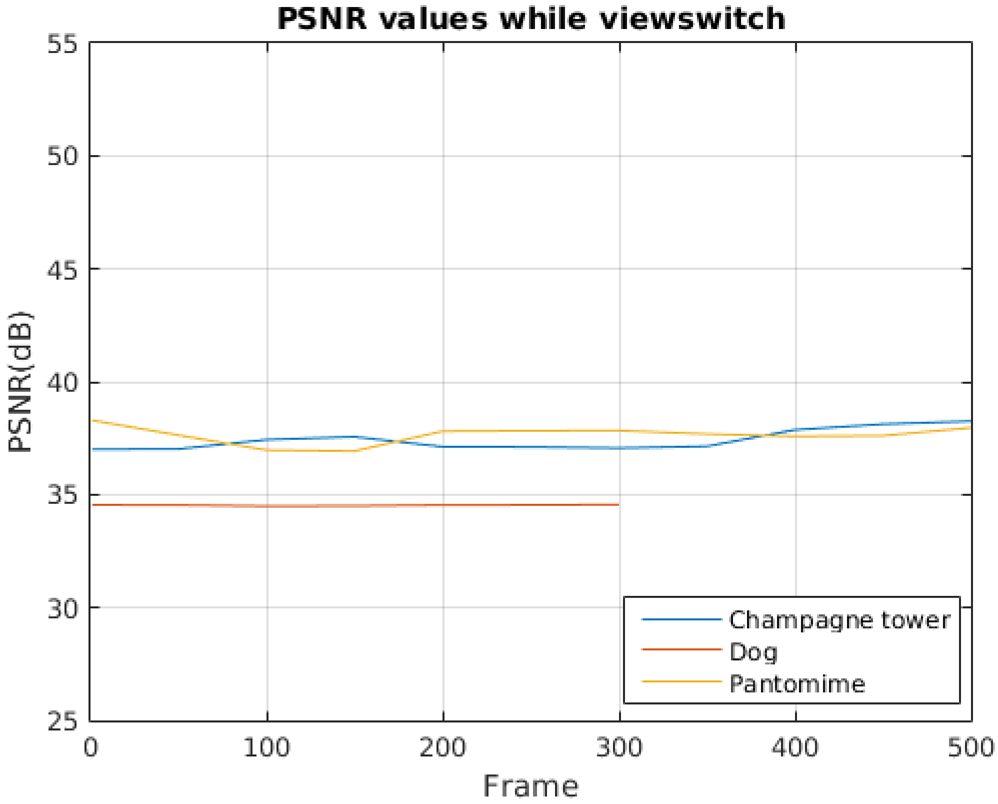


Figure 6.3: PSNR performance on viewchanges

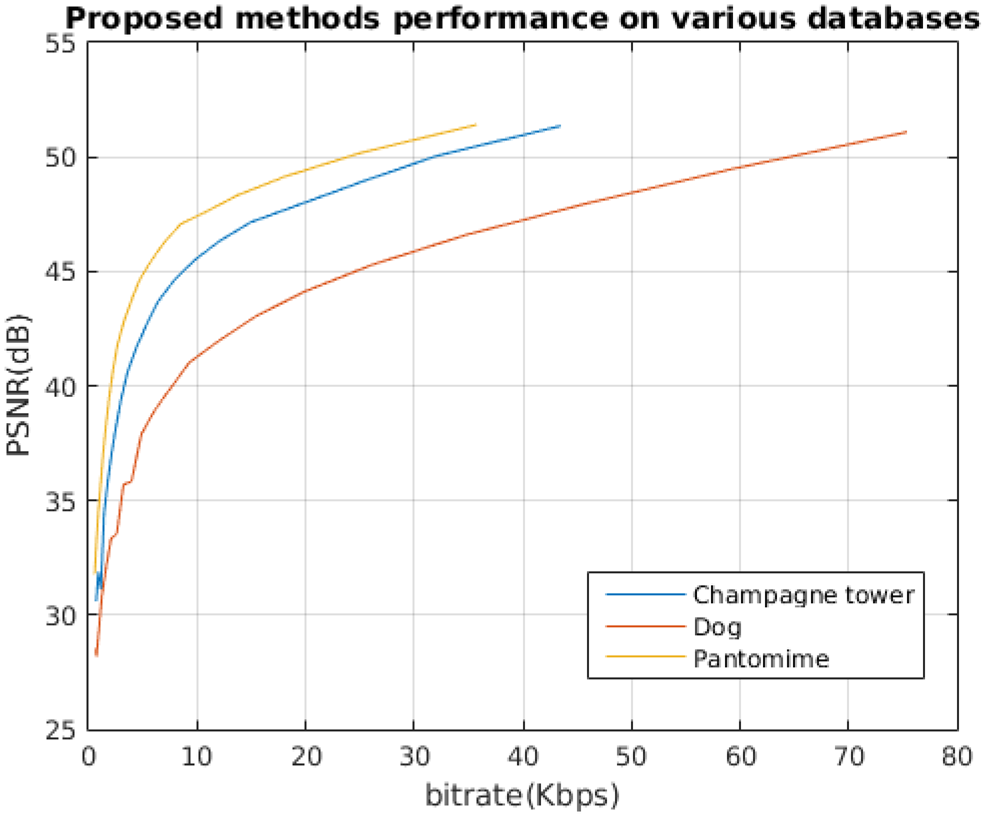


Figure 6.5: PSNR performance of proposed method

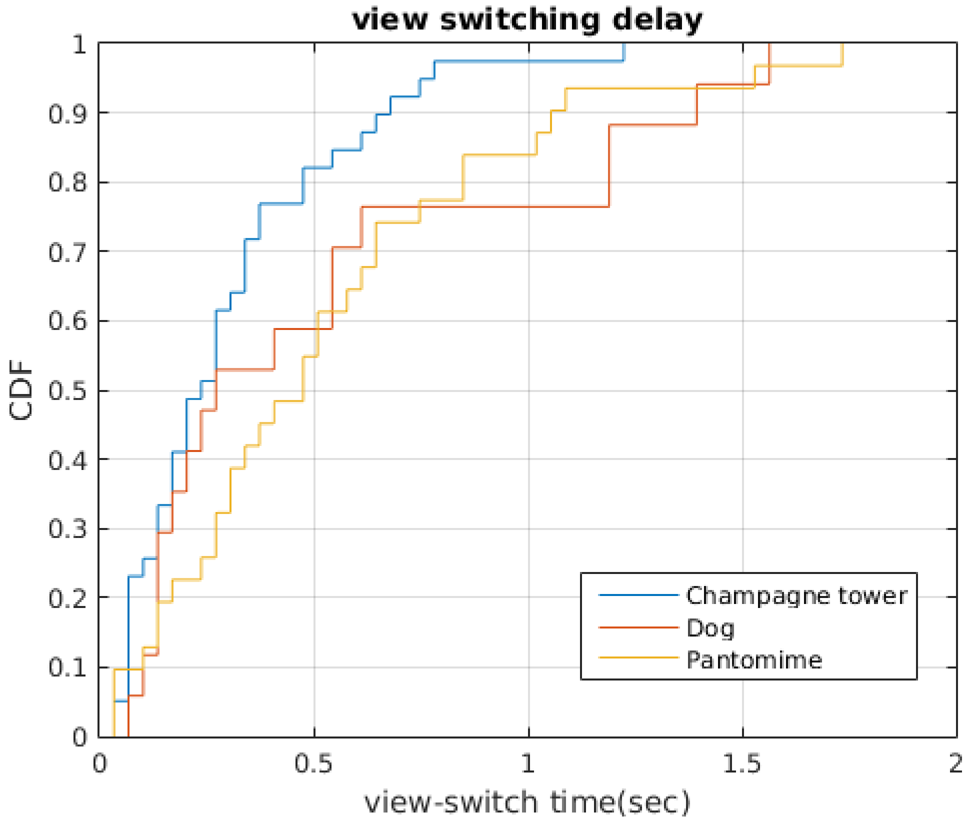


Figure 6.4: CDF of view-switching delay

Figure 6.5 shows the PSNR performance versus bit-rate for the selected three MVV sequences for the proposed method. Each view was encoded using `HM v15.0` reference software for HEVC [41] for compression. The bit-rate has been calculated on the first frame as Figure 6.3 shows almost constant rate of change in PSNR over frames and in terms of bits per frame per camera and converted it to bits per second using frame-rate. In order to provide random access, standard encoder-intra-main configuration has been used with Group of Picture (GOP) 1 and QP from 50 to 12, decrementing 2 to perform the experiment. From the plots, it can be inferred that proposed method provide higher PSNR with bit-rate resulting good quality of video for transmission.

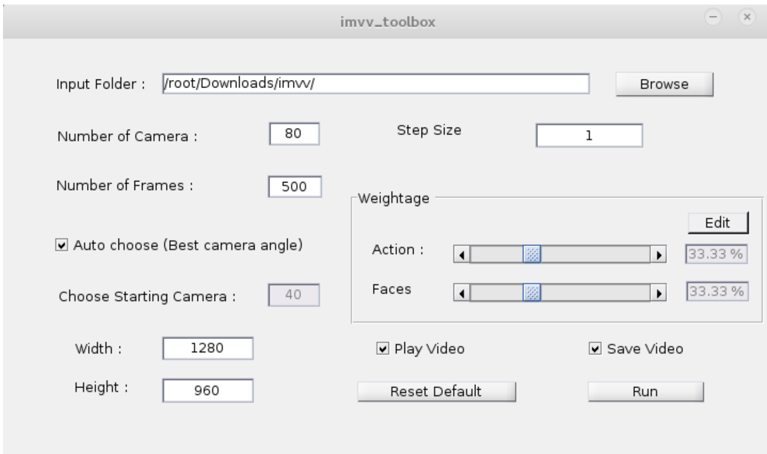
In Table 6.1 a comparison has been performed with [24] for databases *Champagne_tower*, *Dog* and *Pantomime*. Both average and mode of viewswitch time has been shown in order to have better interpretation about time (*sec.*) consumption for switching from current view to predicted view. It has been assumed that users having

Table 6.1: Table of view switch time(sec) comparison

		Reference Method			Proposed Method
		$\mu = 20$	$\mu = 50$	$\mu = 70$	
Average	Champagne tower	0.80	0.29	0.93	0.48
	Dog	0.43	1.23	1.88	0.35
	Pantomime	1.02	0.46	0.72	0.76
Mode	Champagne tower	0.79	0.12	0.79	0.46
	Dog	0.09	1.05	1.75	0.20
	Pantomime	1.63	0.62	0.09	0.10

Laplacian distribution of 400 viewpoints with standard deviation 3 and mean 20 and 50 and 70 respectively. For simplicity, viewpoints are taken equal to camera views. From the plot it can be inferred that proposed method takes less view switch time in average. As changes in mean, which is taken arbitrarily in reference method also changes viewswitch time delay. Experiment shows those increase in viewswitch time delay in Table 6.1.

Also for ease of experiments a toolbox with GUI has been created in MATLAB, as shown in Figure 6.6. Where user can specify, number of frames, or camera as mentioned in flow diagram 5.3. Also user can specify the starting camera, or it will be calculated as per proposed method. And user can provide weightage of action and faces to generate ROI.



imvv_toolbox

Input Folder : /root/Downloads/imvv/

Number of Camera : 80

Step Size 1

Number of Frames : 500

☑ Auto choose (Best camera angle)

Choose Starting Camera : 40

Weightage

Action : 33.33 %

Faces 33.33 %

Width : 1280

Height : 960

☑ Play Video

☑ Save Video

Figure 6.6: GUI for automatic view switch

6.2 Coding

In order to have more concrete result of coding parameter and high compression for bit-stream, various testing on HM-HEVC and MV-HEVCV and 3D-HEVC has been performed. Below table 6.2 provides comparative analysis done on HM-HEVC and MV-HEVC for encoding 320 frames of "Champagne tower" data set of camera "4,8,12" views.

Table 6.2: Table of encoding time comparison

Camera	Nr of Frames	Encoding time(sec)	Decoding time(sec)
4 (HM-HEVC)	320	1205.168	6.597
8 (HM-HEVC)	320	1218.508	6.916
12 (HM-HEVC)	320	1235.014	7.093
4,8,12 (MV-HEVC)	320	22553.853	16.428

It can be observed from the experiment, that the coding tme taken for number of frames are almost similar for techniques like HM-HEVC. Though as expected it should be higher for MV-HEVC as there involved inter prediction among camera views. So computational complexity and time complexity goes higher for those prediction algorithm. Also picking up different cameras can significantly affect the coding complexity.

Further experiment with the coding parameters has been done, where a GUI for auto-generating the configuration script for HM-HEVC/3d-HEVC has been done. As there are numerous coding parameters, which need to change in order to configure perfectly and most of them usually done in command-line as user input or creating a configuration script and pass it in command-line. The GUI has grate advancement for performing quick tests. The example of GUI screen can be seen in Figure 6.7. Rather than transforming command-line to easy GUI, it has following features.

1. Only view or Depth + View options can be selected .
2. Number of frames need to encode can be selected.
3. QP can be selected.

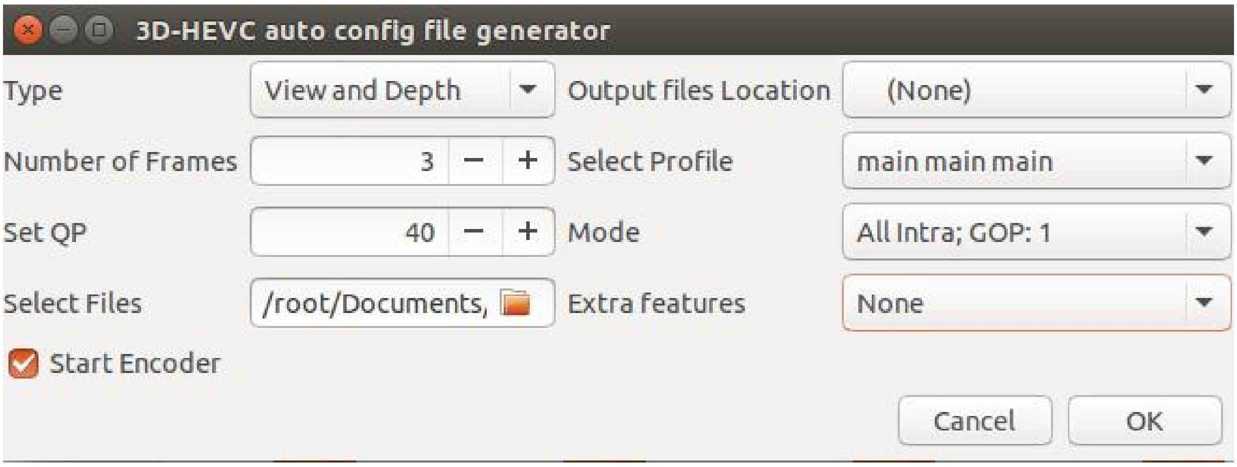
4. Location of the input camera views can be selected, and stored for future use.

5. Location of output bit-stream and configuration script can be selected, and stored for future use.

6. User can choose automatically start encoding or not.

7. Profile can be selected from (main-main-main, main-main-3dmain, main-main-multiview main).

8. GOP (1,4,8) and Mode (All Intra, Low Delay, Random Access) can be selected



3D-HEVC auto config file generator

Type: View and Depth

Number of Frames: 3

Set QP: 40

Select Files: /root/Documents,

Output files Location: (None)

Select Profile: main main main

Mode: All Intra; GOP: 1

Extra features: None

☑ Start Encoder

Figure 6.7: GUI for auto-configure script

Also some complex new feature has been updated for further testing on the coding parameters as follows:

1. It can process multiple stream in parallel, which is not possible in standard encoding script.

2. It can create two separate bit-stream files for even and odd views, in order to transmit bit-stream with separate two channels.

3. It can create individual bit-stream with all intra for multiple views, which is also not possible by standard encoding script.

7 Conclusion and Future Work

7.1 Conclusion

The method demonstrated three principle techniques from dynamic camera allocation based on salience feature and face in video frames, prediction on viewswitch and bitrate allocation for optimised compression. Experiment on contemporary databases have been done and the results are showing the superiority of the proposed method over the state-of-the-art techniques.

7.2 Future Work

The planned future work can be categorised in the below major parts.

7.2.1 View Synthesis

Currently most of the work has been done with the videos having only views without the depth information. Further enhancement of the current method can be done by including view synthesis for intermediate cameras, which is not physically present to capture the object. But using 3D-HEVC we can synthesize the intermediate views. That will create an option for the user to choose any random points among camera locations. Figure 7.1 [10] shows general idea of synthesizing new views. In order to generate new view, both left and right side camera views and their depth information is required, which has also been shown in the mentioned figure.

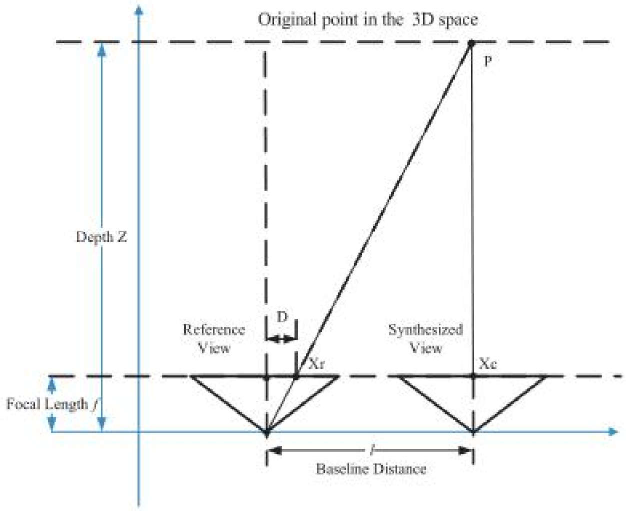


Figure 7.1: Synthesizing of intermediate view from left and right camera information, source : [10]

This view-synthesizing is helpful in generating new views in if we have transmitted view and depth information of both left and right camera view, so there will be no need to send signal to server for new view request, which will eliminate RTT delay.

7.2.2 Prediction analysis

As already discussed previously bit-rate can be controlled by using the camera coding parameters, the target is to use coding parameters to minimize the size of the stream.bit file, which is the output after encoding is done. There are many parameters playing major roles in configuration script. One of them is the prediction mode. Which is majorly categorised as Intra and Inter Prediction modes. In future work, target is to study the behaviours of Inter and Intra prediction mode, and adjust them efficiently to reduce the stream.bit file size. Which will give advantage while transferring data from server to client.

7.2.3 Simulation

Another goal in future work is to test the proposed method using real time network, and analyse the result. The plan is initially to simulate users and networks,

- Single User / Multiple user
- 3G / LTE / High bandwidth networks (fibre optics)

For users both cases are extremal important, but in real scenario cases will be multiple users. But single user test is required to simulate the prediction algorithm and get efficiency over network transmission of that algorithm in prediction point of view. And multiple user can be visualize as multiple single users, where main challenge is transmitting data through many channels. Those views can be similar or different based on the users choice. And effectively distribution of views through various channels can be experimented in this scenario.

Though most of our focus will be testing those scenarios in LTE network, but also with other networks like 3G where bandwidth is much lesser than LTE and also other networks, where virtually much higher bandwidth can be achieved will also be tested to have testing information for next generation networks coming in the network industry.

7.2.4 GUI upgrade

And upgrade the GUI for ease of testing will be done. Few features which will be involved in the GUI upgrade, are as follows.

- Use of different QP in different parallel encoding process
- Cross platform compatible
- More robust parallel process for different sequences, views, depths.

- More robust multiple bit stream output
- Grouping by different views, QPs, sequences etc.

Acknowledgements

I would like to thank my supervisors Gholamreza Anbarjafari and Cagri Ozcinar for all the advice and guidance on writing this thesis.

References

[1] "Fujii.nuee.nagoya-u.ac.jp," 2016, http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/.

[2] R. Suenaga, K. Suzuki, T. Tezuka, M. P. Tehrani, K. Takahashi, and T. Fujii, "A practical implementation of free viewpoint video system for soccer games," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 93 930G–93 930G.

[3] "Cisco visual networking index: Forecast and methodology, 2014-2019 white paper," 2016, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html.

[4] http://cdn.redsharknews.com/images/SD_to_8K_graph.jpg.

[5] 2016, http://www.5kplayer.com/video-music-player/img/uhd-imac-8k-zjy.jpg.

[6] V. Limited, "Vcodex," 2016. [Online]. Available: http://www.vcodex.com/

[7] F. Bossen, D. Flynn, and K. Sühring, "Hevc reference software manual," *JCTVC-D404, Daegu, Korea*, 2011.

[8] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[9] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint tv," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 67–76, 2011.

[10] F. Zou, D. Tian, A. Vetro, H. Sun, O. C. Au, and S. Shimizu, "View synthesis prediction in the 3-d video coding extensions of avc and hevc," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 24, no. 10, pp. 1696–1708, 2014.

[11] J.-G. Lou, H. Cai, and J. Li, "Interactive multiview video delivery based on ip multicast," *Advances in Multimedia*, vol. 2007, 2007.

[12] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Using distributed source coding and depth image based rendering to improve interactive multiview video access," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 597–600.

[13] A. De Abreu, P. Frossard, and F. Pereira, "Optimized MVC prediction structures for interactive multiview video streaming," *Signal Processing Letters, IEEE*, vol. 20, no. 6, pp. 603–606, 2013.

[14] W. Xu, J. Zou, and H. Xiong, "Interactive multiview video scheduling through bargaining," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3590–3594.

[15] G. Lafruit, K. Wegner, and M. Tanimoto, "Final Draft Call for Evidence on FTV," ISO/IEC JTC1/SC29/WG11/, Poland, Warsaw, Tech. Rep. MPEG2015, June 2015.

[16] A. Smolic, "3D video and free viewpoint videoâĂŤFrom capture to display," *Pattern recognition*, vol. 44, no. 9, pp. 1958–1968, 2011.

[17] G. Bang, G. Lafruit, G. s. Lee, and N. H. Hur, "Introduction to 3603D video application and requirements for FTV discussion," ISO/IEC JTC1/SC29/WG11/, Geneva, Switzerland, Tech. Rep. MPEG2015/M37351, Oct. 2015.

[18] H. Huang, B. Zhang, S.-H. G. Chan, G. Cheung, and P. Frossard, "Coding and replication co-design for interactive multiview video streaming," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2791–2795.

[19] G. Cheung, A. Ortega, and N.-M. Cheung, “Interactive streaming of stored multi-view video using redundant frame structures,” *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 744–761, 2011.

[20] E. Kurutepe, M. R. Civanlar *et al.*, “Client-driven selective streaming of multi-view video for interactive 3dtv,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1558–1565, 2007.

[21] T. Scandarolli, R. L. de Queiroz, D. Florencio *et al.*, “Attention-weighted rate allocation in free-viewpoint television,” *Signal Processing Letters, IEEE*, vol. 20, no. 4, pp. 359–362, 2013.

[22] Y.-C. Chen, D.-N. Yang, and W. Liao, “Efficient multi-view 3D video multicast with depth image-based rendering in LTE networks,” in *Global Communications Conference (GLOBECOM), 2013 IEEE, pages=4427–4433*. IEEE, 2013.

[23] C. Fehn, “Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV,” in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.

[24] C. Dorea and R. L. de Queiroz, “General rate-allocation in free-viewpoint television,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 145–149.

[25] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standardsâĂŤincluding high efficiency video coding (hevc),” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1669–1684, 2012.

[26] F. Bossen, B. Bross, K. Suhring, and D. Flynn, “Hevc complexity and implementation analysis,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1685–1696, 2012.

[27] X. Xiu, G. Cheung, and J. Liang, "Frame structure optimization for interactive multiview video streaming with bounded network delay," in *Image Processing (ICIP), 2011 18th IEEE International Conference on.* IEEE, 2011, pp. 593–596.

[28] ——, "Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis," *Multimedia, IEEE Transactions on*, vol. 14, no. 4, pp. 1109–1126, 2012.

[29] C. Zhang and D. Florêncio, "Joint tracking and multiview video compression," in *Visual Communications and Image Processing 2010.* International Society for Optics and Photonics, 2010, pp. 77 440P–77 440P.

[30] Z. Chen, L. Sun, and S. Yang, "Overcoming view switching dynamic in multiview video streaming over p2p network," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010.* IEEE, 2010, pp. 1–4.

[31] T. Scandarolli, R. L. de Queiroz, and D. A. Florencio, "Attention-weighted rate allocation in free-viewpoint television," *Signal Processing Letters, IEEE*, vol. 20, no. 4, pp. 359–362, 2013.

[32] Y.-C. Chen, D.-N. Yang, and W. Liao, "Efficient multi-view 3d video multicast with depth image-based rendering in lte networks," in *Global Communications Conference (GLOBECOM), 2013 IEEE.* IEEE, 2013, pp. 4427–4433.

[33] C. Dorea and R. L. de Queiroz, "General rate-allocation in free-viewpoint television," in *Image Processing (ICIP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 145–149.

[34] H. Huang, B. Zhang, S.-H. G. Chan, G. Cheung, and P. Frossard, "Coding and replication co-design for interactive multiview video streaming," in *INFOCOM, 2012 Proceedings IEEE.* IEEE, 2012, pp. 2791–2795.

[35] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, “Client-driven selective streaming of multiview video for interactive 3dtv,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1558–1565, 2007.

[36] G. Cheung, A. Ortega, and N.-M. Cheung, “Interactive streaming of stored multiview video using redundant frame structures,” *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 744–761, 2011.

[37] D. Walther and C. Koch, “Modeling attention to salient proto-objects,” *Neural networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[38] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[39] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains,” *The annals of mathematical statistics*, pp. 1554–1563, 1966.

[40] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: Evidence and implications,” in *INFOCOM’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 126–134.

[41] Itu.int, “H.265.2(10/14) Reference software for ITU-T H.265 high efficiency video coding,” 2014. [Online]. Available: http://www.itu.int/rec/T-REC-H.265.2

[42] C. De Boor, C. De Boor, C. De Boor, and C. De Boor, *A practical guide to splines*. Springer-Verlag New York, 1978, vol. 27.

Non-exclusive licence to reproduce thesis and make thesis public

I, Suman Sarkar (date of birth: 7th of January 1988),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

DYNAMIC RATE ALLOCATION OF INTERACTIVE MULTIVIEW VIDEO WITH VIEWSWITCH PREDICTION

supervised by Assoc. Prof. Gholamreza Anbarjafari and Dr. Cagri Ozcinar

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu 20.05.2016