Optimized Coding Strategies for Interactive Multiview Video

THÈSE N^{O} 6843 (2015)

PRÉSENTÉE LE 27 NOVEMBRE 2015

À L'ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE TRAITEMENT DES SIGNAUX 4

ET

À L'INSTITUTO SUPERIOR TÉCNICO (IST) DA UNIVERSIDADE DE LISBOA

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE
ET
DOUTORAMENTO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES (PhD)

PAR

Ana Karina DE ABREU GOES

acceptée sur proposition du jury:

Dr J.-M. Odobez, président du jury
Prof. P. Frossard, Prof. F. M. Bernardo Pereira, directeurs de thèse
Prof. G. Cheung, rapporteur
Dr A. Smolic, rapporteur
Prof. D. Atienza, rapporteur

Suisse
2015

To my parents...

Acknowledgements

It has been a long road, a road that I could have not finished without the guidance and support of many people. Thus, I owe my gratitude to all those people who have made this thesis possible and because of whom my Ph.D experience has been one that I will always remember.

My deepest gratitude goes to my advisor, *Prof. Pascal Frossard*. Thank you for your continuous support during my Ph.D, your patience, motivation, and immense knowledge. I am particularly thankful for the encouragement and all the discussions that kept me motivated during all these years. Moreover, for your moral support during the most difficult times, which was what I needed to move on. I would not have been able to finish my Ph.D work without your support and guidance. My learning experience would have never been complete without my co-advisor, *Prof. Fernando Pereira*, who has been always there to provide insightful comments that have helped me to focus my ideas. I am also thankful for your encouragement on the consistency in my writings and for the careful reading and commenting on the countless revisions of submitted manuscripts.

Besides my advisor and co-advisor, I would like to thank the members of my thesis committee: *Prof. Jean-Marc Odobez, Prof. David Atienza, Dr. Aljoscha Smolic* and *Prof. Gene Cheung*, for their insightful comments and valuable suggestions that helped me improve the content of the final version of this thesis. In particular, I want to thank *Prof. Gene Cheung*, with whom I had the opportunity to work closely and from whom I learned so many things. I am also grateful for the partial support of the Fundação para a Ciência e a Tecnologia (FCT) under the grant SFRH/BD/51443/2011.

A big thank goes to all the members of my lab, LTS4, to the ones that saw me arrive and to the ones I'm leaving behind. Many thanks to the great colleagues I have had in these years: *Tamara, Zafer, Vijay, David, Eirina, Nikos, Elif, Luigi, Alhussein, Dorina, Sofia, Laura, Thomas, Xiaowen, Stefano, Mattia, Pinar, Renata* and *Francesca*. In special, I want to thank *Nikos*, the first person I worked with when I arrived in EPFL, for his patience, motivation and friendship. A special thank goes to *Laura* and *Thomas* with whom I worked on joint projects, thank you for your time, enthusiasm and wise advices that made working a great pleasure. I would also like to thank my officemate *Sofia*, who have become a friend ever since we began to share the office. The last months of this thesis work would have been much more difficult without you. Many thanks also to Rosie for all her administrative support.

Acknowledgements

Life in EPFL goes beyond the lab environment. I would like to thank all the people I have met in the LTS corridor in these years, for the personal time we have spent together. In special, thank you Alia for the nice conversations and your Swedish point-of-view. *Laura* for your positive spirit. *Anna* because it always feels nice to talk with you. *Rafael* and *Ricardo* for the nice feeling of being close home everytime I meet any of you. Thank you to all the people that have made special every working day (*Elda, Meri, Sofia, Dorina, Pinar, Alessandra, Pinar, Francesca, Ashkan, Mahdad, Naghmed, Anil, Eleni* and *Leila*). Special thanks go to my Venezuelan friends: *Nohelys*, *Francisco* and *Jacob*, who make me go back in time and make me feel home every time I see them.

And talking about Venezuela, I want to thank the most basic source of my life energy: *my family*, and this needs to be said in Spanish. Gracias *mamá* y *papá* por su amor, trabajo y sacrificios desde siempre, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy. Jamás se los agradeceré lo suficiente. Gracias a *mi hermano*, mi pedacito de familia en Europa, mi mejor amigo, mi eterno cómplice de travesuras. Gracias a mi *Tía María*, mi modelo a seguir desde que era niña, quien me infundió el amor por las matemáticas y las ciencias. Si no fuese por ti no estaría hoy aquí. A *mis abuelos*, quienes no estan hoy con nosotros, pero son una parte esencial de mi y mis triunfos. **¡Gracias!**

Last but no least, thank you *Jonnahtan* for your endless love and support all these years. Thank you for your encouragement at times I did not think I could go on, for always looking at the bright side of everything, for your faith in me. I don't have the words to thank you enough.

Lausanne, 9 November 2015 Ana De Abreu.

Abstract

Since the beginning of multimedia services, in particular video services, with the invention of the television, considerable effort has always been devoted on reproducing the real world. We have witnessed the transition from black-and-white to color television and from a very low image resolution to current Ultra High-Definition Video (UHDV), 7680 × 4320 pixels. The natural next step in improving the realistic experience in multimedia services is *interactive multiview video* (IMV). IMV promises to enable the users to freely navigate through a scene by selecting their preferred viewpoints from any view position for which the corresponding view is generated. A smooth navigation could be achieved with camera views and views synthesized at the decoder. Ultimately, an infinite number of views will be available to the users, providing a very realistic viewing experience with a wide navigation range. However, the large amount of data required for such navigation experience still represents a challenge for the current systems, which implies the need for new efficient coding strategies that permit to save on storage and transmission resources, while preserving interactivity in the navigation.

In this thesis, we focus on the optimization of coding strategies for IMV systems. In particular, we investigate several problems arising with the large amount of data required by IMV and propose different solutions, such as, (i) optimized multiview video prediction structures for interactive multiview video streaming (IMVS), (ii) an optimal layered representation for adaptive multiview video streaming, and (iii) a Lagrangian multiplier search algorithm for Lagrange-based optimization in constrained rate allocation problems.

First, we address the issues related to the coding techniques for IMV in a multiview video plus depth (MVD) scenario, where texture and depth maps are available for view synthesis at the decoder. Current multiview video coding standards efficiently compress images from different camera views capturing the same scene by exploiting the spatial, the temporal and the interview correlations. However, the compressed texture and depth data have typically many interview coding dependencies, which may not suit IMVS systems, where the user typically requests only one view at a time. In this context, we propose an algorithm for the effective selection of the interview prediction structures (PSs) and associated texture and depth quantization parameters (QPs) for IMVS under transmission and storage constraints. These PSs and QPs are selected such that the visual distortion is minimized during navigation at the decoder, given storage and point-to-point transmission rate constraints. Simulation results show that our novel low complexity algorithm has near-optimal compression efficiency while preserving interactivity properties at the decoder, so that it offers an effective encoding solution for IMVS applications.

Then, considering the limited and heterogeneous capabilities of current networks and de-

coding devices, we propose a novel adaptive solution for IMV based on a *layered multiview representation* where camera views are organized into layered subsets to offer different levels of navigation quality depending on the different client constraints. We formulate an optimization problem for the joint selection of the view subsets and their encoding rates. Then, we propose an optimal and a reduced computational complexity greedy algorithms, both based on dynamic programming. Simulation results show the good performance of our novel algorithms compared to a baseline algorithm, proving that an effective IMVS adaptive solution should consider the scene content, the client capabilities and their preferences, in building adaptive systems for multiview navigation.
Finally, we build on the solution proposed in our second problem and present a general solution to rate allocation problems in multiview video. In particular, we propose a new algorithm to find the optimal Lagrange multiplier in a Lagrangian-based rate allocation problem. This algorithm permits to select the optimal subset of coding units (*e.g.*, views in multiview video) and quantization parameter values (QPs) such that the expected distortion among all the units available at the decoder is minimized given a rate budget constraint. We show that, by combining dynamic programming and a Lagrange-based algorithm with an optimal Lagrange multiplier selection, we are able to reduce the complexity of the rate allocation algorithm and to efficiently solve the allocation problem. We show the performance of our proposed algorithm in both multiview and monoview video scenarios and show that the proposed method is able to compete with complex state-of-the-art rate control techniques.
In summary, this thesis addresses important issues for coding multiview video in the design of efficient IMV systems under resource constraints. Our algorithm to select the optimal PS and QPs in a MVD scenario can improve the quality of the rendered views and it can indeed provide new insights for a deeper understanding of specific IMV coding requirements. We show that our algorithm for a layered representation of multiview video provides an effective adaptive streaming solution for IMV systems with users with limited and heterogeneous capabilities. Finally, our proposed Lagrangian-based rate allocation algorithm with an optimized selection of the Lagrange multiplier represents a general contribution that can be used in both multiview video and monoview video scenarios.

Keywords: Interactive multiview video (IMV), multiview video plus depth (MVD), navigation, streaming, view synthesis.

Riassunto

Sin dalla nascita dei servizi multimediali, in particolare quelli video diffusosi con l'invenzione della televisione, notevoli sforzi sono sempre stati dedicati a riprodurre il mondo reale. Prima la televisione da bianco e nero è diventata a colori, e poi la risoluzione da molto bassa è arrivata all'Ultra High-Definition Video (UHDV) - 7680 × 4320 pixel - riproducendo sempre più fedelmente il mondo reale. Il passo successivo per rendere sempre più veritiera l'esperienza offerta dai servizi multimediali è la tecnologia chiamata *interactive multiview video* (IMV). IMV è un sistema di visione a telecamere multiple che permette agli utenti di navigare liberamente in una scena, selezionando la prospettiva preferita da cui guardare la scena tra un'insieme di angolazioni disponibili (viste multiple). Questa navigazione è possibile grazie a differenti telecamere, che acquisiscono la scena da diverse angolazioni o viste, e grazie alla tecnica di view-rendering che permette di generare ulteriori viste virtuali, sintetizzate direttamente dal ricevitore. In definitiva, gli utenti possono osservare la scena da un numero infinito di viste, fornendo un'esperienza di navigazione molto realistica in una scena che può essere molto ampia. Tuttavia, tale navigazione è possibile al prezzo di una vasta quantità di dati, non sempre sostenibile dagli attuali siatemo di video comunicazione. Tali sistemi dunque necessitano di nuove ed efficienti strategie di codifica che consentano di risparmiare risorse di storage e trasmissione, preservando l'interattività nella navigazione.

In questa tesi, ci concentriamo sull'ottimizzazione delle strategie di codifica per i sistemi IMV. In particolare, indaghiamo problemi derivanti dalla grande quantità di dati necessari in sistemi IMV e proponiamo differenti soluzioni, quali ad esempio, (i) un'ottimizzazione delle strutture di predizione per multi-telecamere per lo streaming dei sistemi IMV (IMVS), (ii) una composizione del segnale in strati (o livelli) per lo streaming adattativo del video di multi-telecamere in sistemi IMV, e (iii) un metodo per definire il miglior moltiplicatore di Lagrange in problemi di ottimizzazione di assegnazione del rate basati sui moltoplicatori di Lagrange.

In primo luogo, ci focalizziamo sulle tecniche di codifica per sistemi IMV in uno scenario *multiview video plus depth* (MVD), composto da un numero limitato di viste e dalle corrispondenti mappe di profondità. Ad oggi, le immagini acquisite da diverse telecamere che catturano la stessa scena sono compresse in maniera efficiente da attuali metodi di codifica video, che sfruttano la correlazione della sorgente a livello spaziale, temporale ed anche inter-vista, cioè la correlazione che sussiste tra telecamere vicine. Tuttavia, a causa della correlazione inter-vista, i dati dell'immagine e della mappa di profondità, una volta compressi, hanno tipicamente molti dipendenze con altre viste. Il che significa che altre viste devono essere decodificate prima di poter decodificare la vista corrente. Tale codifica con dipendenze non

risulta ideale per sistemi IMVS, in cui l'utente richiede in genere solo una vista alla volta. In questo contesto, proponiamo un algoritmo per un'efficace selezione delle strutture di previsione (PSs) inter-vista e parametri di quantizzazione (QPs) dell'immagine e della mappa di profondità per sistemi IMVS considerando vincoli di trasmissione e storage. Questi PSs e QPs sono scelti in modo tale che la distorsione visiva sia minimizzata durante la navigazione dell'utente nella scena, dati i vincoli di storage ed il rate di trasferimento dei dati point-to-point. L'algoritmo proposto è a bassa complessità e dai risultati risulta essere un'efficiente tecnica di compressione (quasi ottimale) garantendo l'interattività della scena in sistemi IMVS.

Sempre per sistemi IMV, caratterizzati da le limitate ed eterogenee capacità sia delle reti attuali che dei dispositivi di decodifica, proponiamo inoltre una nuova soluzione adattativa basata sul concetto di *rappresentazione di multi-telecamera a strati* dove le viste sono organizzate in strati (o livelli). Tale rappresentazione offre così differenti livelli di qualità durante la navigazione per differenti condizioni (in termini di canale o dispositivi) degli utenti. Il problema di ottimizzazione è finalizzato a determinare l'allocazione ottima delle viste nei differenti livelli, definendo anche i rispettivi rate di codifica. Per risolvere tale ottimizzazione, proponiamo due algoritmi entrambi basati su *dynamic programming*. Il primo metodo calcola la soluzione ottimale, ottimizzando simultaneamente tutti i livelli, ma ad un prezzo di elevata complessità. Il secondo metodo invece offre una complessità di calcolo ridotta ma ottimizza ogni livello di telecamere singolarmente. I risultati delle simulazioni mostrano le buone prestazioni dei nostri nuovi algoritmi rispetto ad un algoritmo di riferimento. Ciò dimostra che, per offrire servizi di IMVS a buona qualità, un'efficace soluzione deve adattarsi al contenuto della scena, alle funzionalità del cliente ed alle loro preferenze.

La soluzione di questa ottimizzazione adattativa è infine generalizzata nel terzo contributo di questa tesi, dove presentiamo una soluzione generale per problemi di allocazione di rate in sistemi di visione di telecamere multiple. In particolare, si propone un nuovo algoritmo per trovare il moltiplicatore di Lagrange ottimo in un problema di allocazione di rate basato sui moltiplicatori di Lagrange. Il metodo proposto permette di selezionare il sottoinsieme ottimo delle unità di codifica (ad esempio, le viste in sistemi di telecamere multiple) ed i valori dei parametri di quantizzazione (QPs) in modo tale che la distorsione prevista tra tutte le unità disponibili presso il decodificatore sia ridotta al minimo dato un budget limitato di rate. Abbiamo dimostrato che, combinando dynamic programming ed un algoritmo basato sui moltiplicatori di Lagrange con un'ottima selezione del moltiplicatore stesso, siamo in grado di ridurre la complessità dell'algoritmo di allocazione di rate, risolvendo dunque tale problema in maniera efficiente. Le prestazioni del nostro algoritmo proposto in scenari di visione con telecamere multiple o singole dimostrano che il metodo proposto è in grado di competere con le tecniche complesse di ultima generazione per il controllo del rate.

In sintesi, questa tesi affronta argomenti importanti per la codifica di sistemi di visione di telecamere multiple in servizi IMV quando le risorse sono limitate. Il nostro algoritmo per selezionare l'ottimale PS e QPs in uno scenario MVD può migliorare la qualità delle viste virtuali generate al decodificatore e può fornire nuove intuizioni per una più profonda comprensione di specifici requisiti di codifica IMV. Abbiamo dimostrato che il nostro algoritmo per la rappresentazione a livelli multipli di multi-telecamere fornisce una soluzione efficace

per lo streaming adattativo in sistemi IMV con utenti con capacità limitate ed eterogenee. Infine, l'algoritmo di allocazione di rate che proponiamo è un contributo generale che offre una tecnica efficace di ottimizzazione e può essere utilizzato in scenari di telecamere multiple, ma anche in problemi di allocazione di rate in casi di telecamera singola.

Parole chiave: Interactive multiview video (IMV), multiview video plus depth (MVD), navigazione, streaming, viste virtuali.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The advances in image and signal processing, display technologies, coding and transmission techniques coupled with the rapid increase in computing power, computer storage facilities, communication speed, have led to an increase in both content creation and consumer demand of video services. Moreover, due to the increasing use of personal devices capable of reproducing videos, and to the delivery of content over the Internet, viewers can now enjoy any video service on any device, anywhere, and at anytime. Video experience have shifted from a social experience, where groups of viewers gather in front of the house TV, to a more individual experience, where viewers watch their favorite programs on smaller, more personal devices such as tablets and smartphones. More importantly, users are no longer slaves of their TV sets, they can watch the video they want in their own terms. However, interaction is currently limited to occasional user intervention in the time domain (*e.g.*, pause, play, fast forward and rewind), and users are not yet able to choose their own viewing angle in a 3D scene. For example, a sport scene would be much more exciting if the viewer could get the desired viewpoint of his/her favorite player, bringing the user to the center of the play. Similarly, educational videos would benefit as well from more interactivity, since it permits a better understanding of complex structures, *e.g.*, molecules or engines, if the viewer can rotate them by himself.

Given the trends towards a more personal video experience, we believe that the next step for video services is *interactive multiview video* (IMV). In IMV, an array of cameras first capture the same 3D scene from different viewpoints in order to provide the clients with the capability

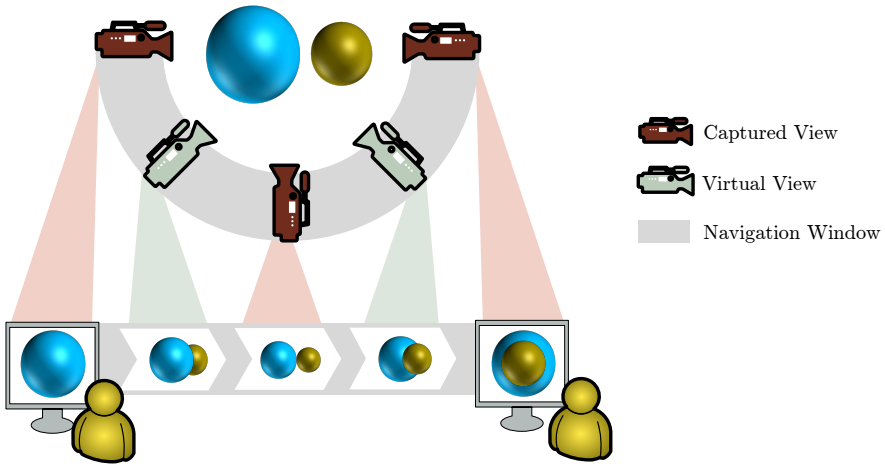


Figure 1.1 – Illustration of an *interactive multiview video* (IMV) where a user can request any view, captured or virtal view, defining the navigation window.

of eventually choosing among different views of the scene. Intermediate virtual views, that are not available from the set of captured views, can also be estimated and rendered at the decoder if relevant information from neighboring views is available. As a result, IMV clients can get the freedom of selecting a viewpoint from a set of captured and virtual views defining an interaction space or a navigation window (Fig. 1.1). These systems have been developing fast in the recent years. Current implementations of IMV systems have considered linear camera arrangements with a few number of camera views. Recent efforts of standardization committees already target Super Multiview Video (SMV) and free navigation [5]; they require more views with a wider and higher-dimensional camera arrangement to have a glass-free 3D experience in the SMV case [6], and a “walking-through" feeling in the free navigation case [7], [8].

IMV however brings important challenges compared to traditional video system, due to the important increase in the volume of data with multiview representations. In IMV many views need to be stored and eventually transmitted to the users. Thus, the design of efficient coding strategies that consider storage and bandwidth resources, complexity, video quality and interaction delay becomes crucial in IMV systems. It is essential to overcome these challenges in order to provide high quality IMV services, which will offer a smooth and high quality navigation experience and lead to a mass adoption of these exciting new applications in the near future.

The coding strategies proposed so far in the literature for multiview video target applications where the full set of captured views are transmitted together to the clients. Given that the different views in multiview video tend to be very correlated, as the different video signals are created from the same scene, current coding solutions exploit the similarities between adjacent views to maximize the compression efficiency, in addition to the redundancies already exploited in traditional monoview video encoders (*i.e.*, temporal and spatial redundancies).

However, in an interactive system only one view is requested at a time. Inter-view coding dependencies unfortunately impose a lot of information that needs to be transmitted in order to decode or render each single view. Providing high view-switching flexibility with reduced coding dependencies can however come at the cost of reducing the compression efficiency. Therefore, there is a need to find effective prediction structures that adapt to the IMV particular needs in trading-off compression efficiency and interaction capabilities.

IMV is an application that requires the transmission of a huge amount of information and therefore it needs a scalable or adaptive solution that ensures that all the clients can enjoy it at the highest possible quality even if they have different bandwidth capabilities. The solutions proposed so far have been an extension of adaptive solutions for traditional monoview video, mainly based on scalable video coding, which allows video to be encoded as a set of 'layers' of increasing quality and complexity. However, the particular characteristics of multiview video have not been exploited for scalable solutions. For instance, instead of transmitting the complete set of views of a multiview video dataset, some views can be omitted from the compressed bit-stream and eventually reconstructed at the receiver side. This solution permits to trade off navigation quality and transmission bandwidth and to adapt the navigation experience to the user capabilities.

In general, efficient compression for single or multiple layers stream raises a rate allocation problem as storage capacity and transmission rate are generally constrained resources. The problem is to find the optimal rate distribution among coding units, *e.g.*, frames in a monoview video or views in a multiview video, such that the quality is optimized given a rate constraint. To solve rate allocation problems, methods based on dynamic programming have been suggested. However, due to the computational complexity of dynamic programing approaches, rate allocation problems are usually solved considering a Lagrangian cost function, where an unconstrained formulation of the problem is possible by using a Lagrange multiplier. However, the search of the optimal Lagrange multiplier is usually overlooked, which jeopardizes the algorithms for rate allocation in practice. Moreover, most of the works tackling the problem of rate allocation focus on traditional monoview video settings and rate allocation problems for multiview video have received only very limited attention.

In this thesis, we address the limitations of current multiview systems and propose novel coding strategies for IMV. We first propose an optimization algorithm to select optimal prediction structures (PSs) for *interactive multiview video streaming* (IMVS). These PSs result from a trade off between compression efficiency and interaction flexibility. Then, a bandwidth-adaptive solution for IMV is presented, where some views can be skipped for encoding/transmission and reconstructed at the user side. Finally, we present a novel algorithm to find the optimal Lagrange multiplier in Lagrangian-based optimization for rate allocation problems. This solution is general enough to be applied on both multiview video and traditional monoview videos.

1.2 Thesis Outline

The outline of this thesis is as follows.

In Chapter 2, we provide an overview of the existing works related to coding strategies for IMV applications, as this is the focus of this thesis. First, we describe the different components of an IMV system and their impact on the coding solutions. Then, we overview the existing coding strategies for multiview video data, considering compression, scalability and rate control properties, along with specific coding requirements for IMV.

In Chapter 3, we investigate the problem of finding the optimal multiview video plus depth prediction structures (PSs) to trade off compression and interaction flexibility in an IMV scenario. We propose a greedy algorithm to find the optimal PS and quantization parameters (QPs) for the texture and depth maps, in a system where the point-to-point transmission bandwidth and the storage capacity are scarce resources. Experimental results show that our new generic algorithm is able to identify a near-optimal PS in the sense of minimizing the distortion while trading off the transmission and storage costs. At the same time, our PS and associated QPs selection algorithm leads to a complexity reduction up to 72% compared to an optimization performed with exhaustive search approach. The research work related to this chapter has resulted in the following publications:

- A. De Abreu, P. Frossard and F. Pereira; *"Optimized Multiview video Plus Depth Prediction Structures for Interactive Multiview Video Streaming"*, IEEE Journal of Selected Topics in Signal Processing, vol 9, no. 3, pp. 487-500, April 2015.
- A. De Abreu, P. Frossard and F. Pereira; *"Fast MVC Prediction Structure Selection for Interactive Multiview Video Streaming"*, Picture Coding Symposium (PCS); San Jose, CA, US; 8-13 December 2013.
- A. De Abreu, P. Frossard and F. Pereira; *"Optimized MVC Prediction Structures for Interactive Multiview Video Streaming"*, IEEE Signal Processing Letters, vol. 20, no. 6, pp. 603-606, June 2013.

In Chapter 4, we consider the scenario where resources constraints prevent the transmission of all the views to all the clients in an IMV system. We propose an adaptive or scalable representation strategy for interactive multiview video streaming (IMVS) systems that adapts to the capabilities of the different clients. This adaptation is performed by varying the number of camera views transmitted to the decoder, hence the navigation quality, according to users capabilities. We consider the problem of jointly determining which views to transmit and at what encoding rate, such that the expected rendering quality in the navigation window is maximized under relevant resources constraints. Simulation results show the benefits of the proposed solution compared to a baseline view selection algorithm. The research work related to this chapter has resulted in the following publications:

- A. De Abreu, L. Toni, N. Thomos, T. Maugey, F. Pereira and P. Frossard; *"Optimal Layered Representation for Adaptive Interactive Multiview Video Streaming"*, Journal of Visual Communication and Image Representation, accepted under minor revision.
- A. De Abreu, L. Toni, T. Thomas, N. Thomos, P. Frossard, F. Pereira; *"Multiview Video Representations for Quality-Scalable Navigation"*, IEEE Conference on Visual Communications and Image Processing (VCIP); Valeta, Malta; 7-10 Dec. 2014.

Then, in Chapter 5, we tackle generic rate allocation problems, where a rate budget should be optimally distributed among the views in a multiview video. We are particularly interested in finding the optimal Lagrange multiplier value when a rate allocation problem is solved by a Lagrangian optimization. We propose a new and effective algorithm for picking the optimal value of the Lagrange multiplier. We illustrate the performance of this algorithm on multiview video data and on traditional monoview video. To appreciate the performance of our rate allocation algorithm we also compare our results to rate control solutions adopted in the reference softwares of current monoview and multiview video standards, namely HEVC [9] and 3D-HEVC [10], showing that a simple strategy as the one proposed compares favorably to more complex rate control solutions. This work is in preparation for publication:

- A. De Abreu, G.Cheung , P. Frossard, F. Pereira; *"Optimal Lagrange Multiplier Values for Constrained Rate Allocation Problems"*, in preparation.

Finally, some conclusions and directions for future work in the field of IMV are presented in Chapter 6.

1.3 Thesis Contributions

The main contributions of this thesis are summarized as follows:

- We propose a greedy algorithm to find the optimal interview prediction structures and quantization parameters (QPs) for texture and depth maps coding in interactive multiview video systems. The optimal PS and QPs minimize the distortion when the point-to-point transmission bandwidth and the storage capacity are scarce resources. Our PS and associated QPs selection algorithm shows a close to optimal performance, leading to a complexity reduction of up to 72% compared to an exhaustive search approach.
- We propose a new type of scalability for IMV compared to classical video, where, instead of transmitting the complete set of views of the multiview video dataset, some views can be omitted from the compressed bit-stream and eventually reconstructed at the receiver side using DIBR methods.

- We investigate the problem of jointly determining the optimal arrangement of views in layers along with the coding rate of the views, such that the expected rendering quality is maximized in a given navigation window. We show that this combinatorial optimization problem is NP-hard, meaning that it is computationally difficult and there are no known algorithms that optimally solves the problem in polynomial time.
- In the framework of an adaptive solution for IMV, we propose a globally optimal solution and, due to its high complexity, a greedy algorithm both based on dynamic programing. The results show that our greedy algorithm achieves a close-to-optimal performance in terms of total expected distortion.
- We propose a generic algorithm that finds the optimal Lagrange multiplier value with a minimum number of iterations given a rate allocation problem to find the optimal subset of coding units and QPs such that the expected distortion among all the available units at the users is minimized.
- We show that by combining dynamic programming and a Lagrangian-based algorithm with an optimal search of the Lagrange multipliers we are able to reduce the complexity of dynamic programing-based algorithms efficiently solving the rate allocation problem.

2

State of the Art

2.1 Overview

Interactive multiview video (IMV) enables the "walking-through" or "navigation" experience by providing multiple views of the same 3D scene. It promises a realistic experience, as users can switch to any viewpoint in order to have a new viewing angle of the same scene. However, this comes at a price. Multiview videos are clearly much larger in size than traditional monoview video and effective coding strategies adapted to the particular needs of the interaction application are needed in order to have a mass acceptance of this technology.

We provide in this chapter an overview of the different modules of interactive multiview video systems [11][12], with a special focus on coding strategies that is the main topic of this thesis. The coding strategies for IMV largely depend on the constraints imposed by the system, *e.g.*, data representation, the bandwidth of the transmission channel, storage capacity limitations and the type of service offered to the user. Therefore, we first describe the different components of the IMV system in Section 2.2 and their impact on the selection of the coding strategy. Then, in Section 2.3 we overview the existing coding strategies for multiview video data, along with specific target applications.

2.2 Interactive Multiview Video System

In an IMV system, users are able to freely navigate within a scene by choosing their own viewing angle. This can be achieved by capturing the scene by a set of calibrated cameras

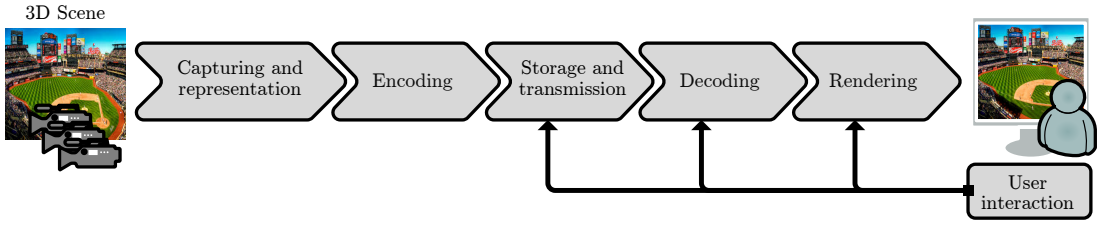


Figure 2.1 – Interactive multiview video (IMV) system. Its components can be classified into: capturing and data representation, coding, storage and transmission, rendering and user interaction.

from different viewpoints. These synchronized video signals can be transformed into a data representation that eventually permits the synthesis or rendering of intermediate virtual views. Then, this acquired and/or transformed data is encoded and eventually transmitted when requested. In general, the components of an IMV system can be classified into five parts: capturing and data representation, coding, storage and transmission, rendering and user interaction. The processing chain of an IMV system is illustrated in Fig. 4.1.

Capturing and Data Representation

Interactive multiview video navigation requires multiple views representing the same scene from different angles. These views can be captured by an array of cameras or they can be computer generated. Different multi-camera setups have been used for capturing the same scene [13] [14], such as 1-D or 2-D linear arrays and 1-D arc arrangements. The camera set-up has an impact on the entire IMV processing chain. For instance, it defines the navigation range offered to the users and determines the quality of the rendered or synthesized views, as this quality usually depends on the camera spacing, camera resolution and the distance from the cameras to the scene. The acquisition of multiview data can be determined by the multiview data representation or format required by the IMV system. Basically, two types of multiview data representation can be distinguished [15], given an IMV application: multiview video (MVV) and multiview video plus depth (MVD). The former is obtained by acquiring texture information from a set of synchronized cameras. Figure 2.2 illustrates the MVV data representation. The latter refers to a data representation where, for each captured frame,

Figure 2.2 – Multiview video (MVV) data representation for *Shark* dataset, provided by NICT for MPEG FTV standardization [1]. Views 20, 40 and 182 (from left to right) and the corresponding first frame in the time domain are used as illustration.

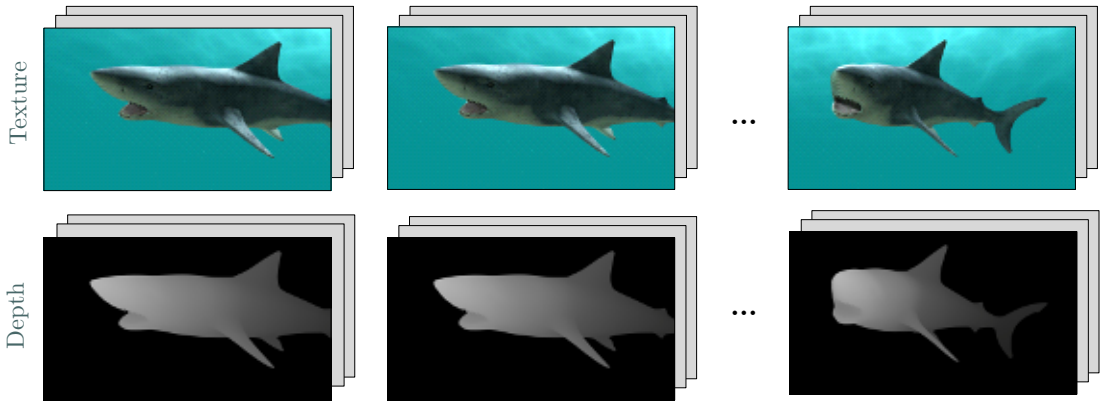


Figure 2.3 – Multiview video plus depth (MVD) data representation for *Shark* dataset, provided by NICT for MPEG FTV standardization [1]. Texture and depth maps of views 20, 40 and 182 (from left to right) and the corresponding first frame in the time domain are used as illustration.

there is an associated depth map, which is eventually used for intermediate view rendering purposes. Depth maps are gray-scale images where each pixel represents the distance between the camera and its position in the 3D scene. Depth maps can be obtained using special camera sensors [16] or they can be estimated from images captured by two or more cameras by solving for stereo correspondences [17]. Figure 2.3 illustrates MVD data representation. The type of multiview data representation used defines the coding strategy that should be followed. This is further explained in Section 2.3.1.

Storage and Transmission

In IMV a user watching a particular view can send at any time a request to a server to switch to a different view while continuing temporal playback. In general, the full set of encoded captured views can be stored in a unique video server [18] [19][20] or the content can be replicated in different local servers closer to the clients to reduce network congestion risks and delays [21]. Then, two main types of transmission models can be identified for IMV. One where the full set of encoded views is transmitted to the users and another one where only the data that users need to decode or render the requested view is sent. The former allows the user to get the requested data from its own memory and decode or render the requested view saving in interaction delay, as all the views have been previously transmitted to the user. In this case, a coding approach that takes into account all the inter-view redundancies is important in order to maximize the compression efficiency, due to the huge amount of data that needs to be sent. However, transmitting the entire multiview video dataset when the user only requests one view at a time can be very demanding in terms of bandwidth and it is not commonly adopted in IMV. The latter transmission model, usually called *interactive multiview video streaming* (IMVS) [18] [20], can potentially reduce bandwidth utilization since only the data required to decode or render the user requested view is transmitted. In this case, a coding strategy that

does not exploit many inter-view coding dependencies among the full set of encoded views would be preferred, as less additional information needs to be transmitted in order to decode and render a requested view.

Rendering

The number of camera views that can be stored and transmitted over the network are limited by the resources available in the system. Therefore, in order to provide a wide and smooth navigation range, view synthesis needs to be used at the decoder for reconstructing novel views between the reference camera views. Rendering techniques have received a lot of attention in the literature. One of the first approaches used for generating virtual views is the model-based rendering (MBR). In MBR, 3D models of the scene are used to create new virtual views [22] [23]. The cost of MBR techniques is high and it depends on the complexity of the scene. Moreover, generating realistic virtual views with MBR involves expensive computation. Therefore, image-based-rendering (IBR) methods [24] [25] have been developed more recently, where the acquired camera images are the main information used for synthesizing new viewpoints. Differently from MBR methods, the rendering cost with IBR is independent of the scene complexity and the texture mapping fidelity of the generated virtual views does not depend on complex 3D models but rather on the available views. IBR techniques have been classified into three categories [26], namely: i) rendering with no geometry; ii) rendering with implicit geometry; and iii) rendering with explicit geometry. The more the geometric information about the scene, the less the number of camera views that are needed for synthesis. Among the different IBR methods, we can find the light-field rendering method [27], which uses many images in order to synthesize a new viewpoint without considering any geometric information. Then, view interpolation [28] is an example of a rendering method that uses implicit geometric information to render new viewpoints by interpolating, generating new views by interpolating the optical flow between corresponding image points. Finally, texture-plus-depth or *depth image-based rendering* (DIBR) methods [17] [29] [30][31] use geometric information of the scene, in particular per-pixel depth information, in order to synthesize novel viewpoints with a limited number of reference images. In DIBR, pixels from a reference image are projected into the 3D world, using the respective depth data and camera parameters, and then these 3D points are projected back into the image plane of a "virtual" camera, which is located at the required viewing position, this is also called 3D-warping [29]. In this process, some pixels that are occluded in the original texture view can become visible in the virtual view (*i.e.*, disocclusions), meaning that no texture information is available for these pixels. As a result, in order to reduce the occurrence of disocclusions, the unknown pixels for a first reference view can be filled with the projected information from a second reference view when available, through a blending process (Fig. 2.4); otherwise, inpainting [32] methods can be used. Dealing efficiently with occlusions and disocclusions in the synthesized views remains a challenge [15].

In general for IBR techniques and a given navigation range, the quality of the synthetic views

increases with the number of available camera views at the decoder and their encoded quality. This becomes a common trade-off in the design of IMV systems between the number and quality of camera views and the view synthesis quality. Thus, the encoding strategy becomes essential in selecting the number of views to encode and the allocation of rate or quality among the selected views, in order to maximized the quality of both the encoded and the synthesized views given some network resource constraints.

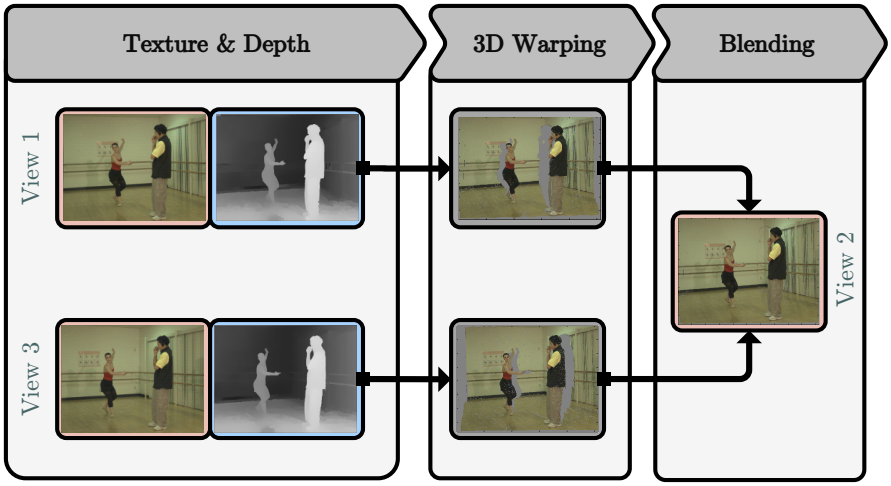


Figure 2.4 – DIBR method using view 1 and view 3 of *Ballet* dataset [2] to generate view 2. First step is the *3D warping*, where pixels from the right and left reference views are projected into the virtual view position. Then, both projections are merged in a *blending* process to fill in disoccluded pixels.

User Interaction

Users profit from IMV by periodically requesting view switches to navigate through the 3D scene. They can manually interact with the system via a traditional keypad [19] or through head motion [33], [34]. Mainly, two types of user interactions can be identified: *random access* and *view-switching*. In random access, view-switches occur from/to any viewpoint in the multiview set at the same time instant. If view-switches occur only through adjacent frames, then a frozen time effect [19] is obtained where the scene is frozen in time and the camera rotates/translates around the captured scene. In the view-switching interaction mode, users are able to switch flexibly from one camera view to another as the video continues along time. Usually, in the view-switching mode, view switches occur through adjacent views for a smooth navigation. The user interaction mode has an impact on the different modules of the IMV processing chain, in particular on the coding solution. Random access is guaranteed by using independently encoded frames, as view-switches occur at a fixed time instant, while in view-switching different coding structures are needed, what makes independent coding inefficient in terms of compression efficiency [18], [35] and [36].

2.3 Coding for Interactive Multiview Video

Uncompressed multiview video signals accounts for a huge amount of data, thus efficient *coding* techniques are essential for enjoying such applications [37]. Moreover, due to the heterogeneity of transmission scenarios and general limited transmission rate, *scalability* of the coded bitstream and *rate control* solutions are also very desirable features in any video coding solution [38]. In this section, we first provide an overview of the different multiview video coding strategies targeting the compression, scalability and rate control issues. Then, we describe particular solutions to be considered for IMV when the users interact with the multiview content.

2.3.1 Multiview Video Coding

The simplest way to encode multiview video is to independently encode each view using a monoview video coding standard, *e.g.*, the H.264/MPEG4 advanced video coding (AVC) standard [39] or the High-Efficiency Video Coding (HEVC) [40] standard, and to independently transmit each view when requested. This is usually referred as *simulcast* [41]. The advantage of this coding technique for IMV, apart from its simplicity, is that only the requested view needs to be transmitted to the client, which minimizes network resources and computational complexity at the decoder. Moreover, it guarantees compatibility with state-of-the-art monoview codecs. However, the different views in multiview video tend to be very correlated (because of the spatial proximity of the capturing cameras) and this coding scheme does not exploit the inter-view redundancy of this type of content, which makes it sub-optimal in terms of the overall rate-distortion trade-off, consuming a lot of storage resources. Therefore, multiview video coding solutions considering the inter-view correlation of this type of data have been proposed. In general, different solutions are proposed depending on the type of data representation they use, namely MVV or MVD.

Given a multiview video (MVV) data representation (Fig. 2.2), multiview video coding schemes target the efficient compression of multiple views capturing the same video scene by enabling inter-view prediction to improve compression capability, as well as supporting traditional temporal (inter-frame prediction) and spatial prediction (intra-picture prediction). This allows frames from adjacent views to be used for prediction of a frame in a current view and at the same time instant. An extension of the H.264/MPEG4-AVC monoview video coding standard, referred as Multiview Video Coding (MVC) standard [42] [1], and a multiview extension of the HEVC standard, denoted as MV-HEVC [43] have been proposed to encode this type of content. They ensure compatibility with the corresponding monoview video coding standards, meaning that one of the views can be independently decoded by a monoview decoder, such as H.264/MPEG4-AVC or HEVC. Figure 2.5 illustrates two common prediction structures used by different coding schemes for MVV, denoted here as IBP and IP, as these are the types of

[1] In this thesis when referring to the Multiview Video Coding standard the acronym MVC is used, otherwise multiview video coding refers to a general codec for multiview video

frames used to encode the anchor frames (*i.e.*, frames that do not use temporal prediction for encoding, although they do allow interview prediction from other views in the same time instant [44]) of the different views. In general, the coding schemes for MVV preserve the encoded video quality, as novel views are not synthesized at the decoder, avoiding the complication associated to view synthesis.

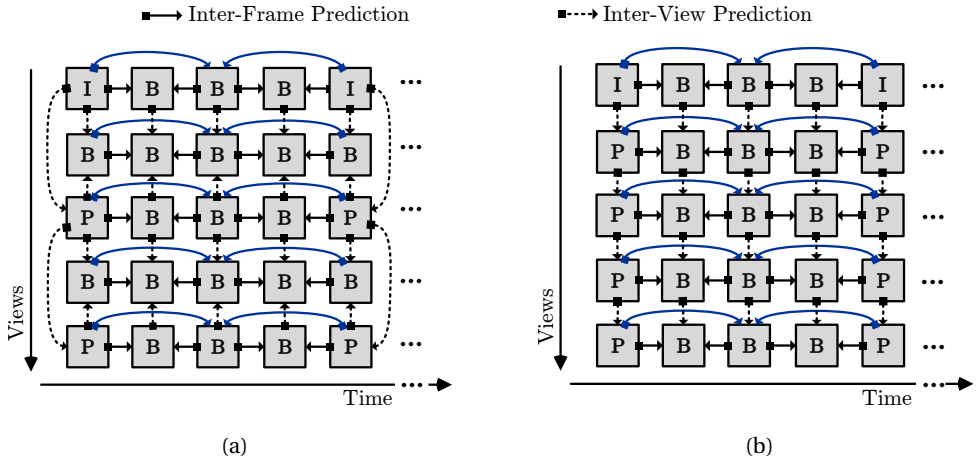


Figure 2.5 – Inter-frame and inter-view prediction structures commonly used in MVC and MV-HEVC standards. (a) IBP and (b) IP prediction structures. Hierarchical B prediction structure is used in the temporal domain.

When the multiview video plus depth (MVD) format is used to represent the multiview data (Fig. 2.3), texture and depth maps need to be efficiently encoded to enable high quality view synthesis at the decoder. Initial results have shown that the depth maps could be efficiently compressed using standard texture coding algorithms [45], where MVC and MV-HEVC standards have been used to encode texture and depth information as separate bit-streams. However, such coding schemes do not exploit the similarities of the motion information of the texture and the depth of the multiview video. Moreover, depth maps have characteristics that are different from texture data (*e.g.*, large monochromatic areas and sharp edges), thus it requires specific coding methods. In order to maximize the compression efficiency, new MVD coding schemes have been proposed where joint coding of depth and texture information is considered, allowing both inter-view prediction for texture and depth maps and inter-component prediction between texture and depth data. Multiview video coding standards as the 3D extension of HEVC (3D-HEVC) [46] have been proposed to this end, offering new coding modes for depth maps using view synthesis prediction and optimization at the encoder. The MVD coding models are illustrated in Fig. 2.6, for the texture and depth components of five views. For the sake of simplicity, the temporal inter-frame prediction is not considered in the figure. A straightforward advantage of these coding schemes is that they provide explicit depth information that could be used for view synthesis at the decoder. However, high-quality view synthesis requires precise depth information [47], which poses challenges on depth coding algorithms.

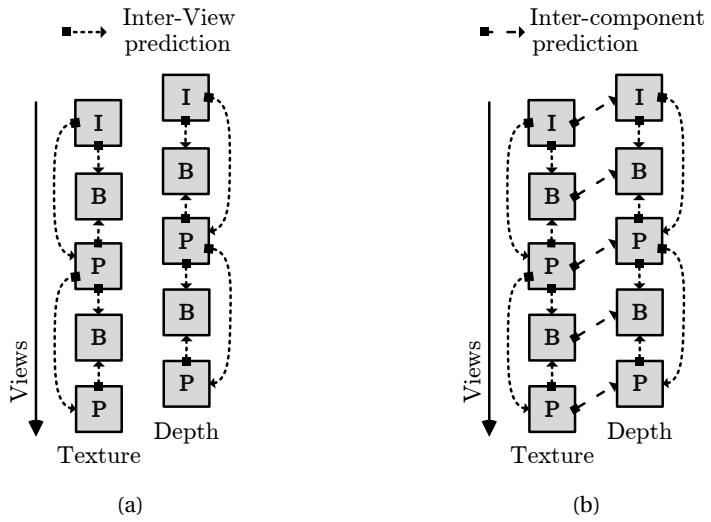


Figure 2.6 – Prediction structures for MVD. (a) Inter-view only prediction (texture and depth maps are independently encoded). (b) Inter-view and inter-component prediction (depth maps are predicted using auxiliary information associated to the texture data).

The efficient compression of multiview videos, and video in general, raises the *rate control* problem when coding and/or transmission bandwidth are constrained. In rate control, the problem is to find the optimal rate allocation among views and/or frames in a (multiview) video sequence. In this context, the term *unit* is used as a general term for frames in traditional monoview video or views in multiview video. These type of problems have a high complexity, in particular for multiview video, due to the search among all possible QPs and total number of units. In the literature, rate allocation problems are usually solved by considering an unconstrained problem based on Lagrangian optimization [48] [49] [50] [51] [52] [53], where a Lagrange multiplier λ is used to define a Lagrangian cost function in the form $D + \lambda R$, as it allows to set different R-D trade-off points by modifying λ. For instance, minimizing a Lagrangian cost function with $\lambda = 0$ is equivalent to minimizing the distortion. On the other way around, minimizing a Lagrangian cost function with a large λ value is equivalent to minimizing the rate. Thus, for each λ value there is an optimal rate allocation solution, meaning that the optimal λ also needs to be found. Usually, search for the best λ is done by swapping its value from an initial lower bound to an upper bound from a predetermined set of λ values [52], where a bisection search can be used to reduce the number of iterations. However, finding an optimal λ is not guaranteed as it depends on the granularity of the search space.

Most of rate allocation algorithms for rate control problems have been proposed for traditional monoview videos [50] [51] [52] [53]. Recently, some effort have been made to propose rate allocation solutions for multiview video. In [48], Kim et al. a trellis-based optimization approach is presented for multiview video where views are predictively coded. The authors do not consider a system where missing views can be synthesized using both texture and depth

maps, thus only the QPs of the texture information are optimized. On the other hand, the work in [49] tackles the bit allocation problem for both texture and depth maps such that the distortion of the encoded and the synthesized views is minimized. The authors optimize both the set of coded views and their QPs, also adopting a trellis-based solution. However, both works, [48] and [49], use a Lagrange cost function where no constructive algorithm for the search of the optimal λ is proposed.

As an alternative to Lagrangian optimization, rate allocation problems can be solved as dynamic programming problems [54] providing optimal solutions; however the complexity is rather high as it is a way of optimally considering all the possible solutions.

Scalability is also a desired feature in a multiview video coding strategy. In current networks, where users may have different access link bandwidth capabilities, it becomes a challenge to offer the same IMV service to all the users, even if the most efficient compression scheme is used. In this context, it becomes important to devise adaptive compression and transmission strategies for IMV systems that adapt to the capabilities of the users. Here, we define scalability as the ability of the decoder to access part of the entire bitstream and still being able to enjoy the IMV experience, even at a reduced quality. The problem of heterogeneous users has been mostly tackled in the literature via *scalable multiview video coding*. For instance, some extensions of the H.264/SVC standard [55] for traditional 2D video have been proposed in the literature for multiview video [56] [57]. In [58], [59] and [60], the authors propose a joint view and rate adaptation solution for heterogeneous users. Their solution is based on a wavelet multiview image codec that produces a scalable bit-stream from which different subsets can be extracted and decoded at various bitrates in order to satisfy different users bandwidth capabilities.

The focus of recent works and standardization activities in the area of multiview video coding has been mainly on compression efficiency, where usually one view is encoded independently and the other views are predictively encoded to maximize the redundancy reduction. However, in IMVS systems where only one of the views is requested at a time, this may not be the most efficient prediction model, as it leads to large transmission costs. For instance, from Fig. 2.5, if the last view in the coding set is requested by a user, all the previous views need to be transmitted first in order to decode the desired view. In the following, we present some coding solutions for IMV applications.

2.3.2 Multiview Video Coding in IMV

Interactive multiview video is characterized by view-switches from/to any viewpoint in the multiview set, virtual or coded. This means that, without any previous knowledge of the navigation path followed by the user, video coding solutions for IMV need to provide flexible viewpoint switching in order to minimize network resource consumption and decoding complexity. A common challenge in IMV systems is to consider both, a solution that exploits the inter-view correlation for efficient coding, and a solution that provides interaction flexibility

to the users minimizing transmission costs and interaction delay.

Recently, some prediction structure (PS) selection algorithms to encode multiview video have been proposed for IMV systems, with the goal of providing multiview video with flexible viewpoint switching by trading off the transmission rate, storage capacity and/or latency. To save transmission bandwidth, different interview prediction structures are proposed in [61] to code various versions of a multiview set in order to satisfy different RD performances. However, this approach brings a high storage cost at the server, as its gain depends on the number of PSs used to encode the multiview sequence. Similarly, the authors in [19] consider the trade-off between flexibility, latency, and bandwidth when proposing three prediction structures in order to offer three different types of interactive experiences to the users. A low-delay random accessibility, as well as low-transmission bandwidth cost is proposed in [62], where a group of GOPs (GoGOP) concept is introduced with interview prediction restricted to the views in the same GoGOP. However, this solution leads to limited compression efficiency. In [63], a *user dependent multiview video streaming for Multi-users* (UMSM) system has been proposed to reduce the transmission rate, where overlapping frames (potentially requested by two or more users) are encoded together and transmitted using multicast, while the non-overlapping frames are transmitted to each user by unicast. This approach is only useful when several users are watching the same video at the same time instant. In addition, if in [63] a random interactivity model is assumed, where a user can switch to any viewpoint, UMSM must transmit all the views to each user, which results in large bandwidth usage. Differently, in [18],[35] and [36], the authors have studied the PSs that facilitate a continuous view-switching by trading off the transmission rate and the storage capacity. The authors have considered a coding system with redundant P- and DSC-frames (distributed source coding), which is unfortunately not compliant with standard decoders. A different approach is followed in [49] and [58] where, given a rate constraint, a set of views is optimally selected at the sender side for encoding. Then, the set of views are transmitted to the users from where they may select an encoded view or reference vieews to synthesize a desired viewpoint.

In addition, some interactive navigation systems have been proposed where the redundancies are not only reduced through coding techniques but also in the data representation adopted. In [64] the authors propose a novel solution where the available navigation domain is partitioned into segments. Each of the segments is then described by a unique reference image (texture and depth maps) and some auxiliary information. This auxiliary information and the reference image allow the user to synthesize any viewpoint in each particular segment, and additional data is only required if the user “moves” to a different segment.

The work of this thesis continues the research efforts in designing a coding strategy that balance compression efficiency and interaction flexibility in the context of multiview video systems for interactive navigation.

3

Optimizing Multiview Video plus Depth Prediction Structures for IMVS

3.1 Introduction

In this chapter, our main goal is to find the optimal interview prediction structure (PS) and associated texture and depth quantization parameters (QPs) to encode a set of views in the context of interactive multiview video streaming (IMVS). In IMVS, the users periodically request view switches and only the data required to decode or render the requested view is transmitted by the server. Efficiently encoding this type of content a priori and without knowing the actual path each user will follow in his/her interactive navigation is a challenging task. Most of recent works and standardization activities have focused on exhaustively exploiting the inherent correlation among the views to improve the overall compression efficiency [45] [65], without considering the penalty in transmission rate that it brings to IMVS systems, as data that is not requested but required for decoding needs often to be transmitted. The coding and the prediction structures for IMVS applications have to be different from other non-interactive multiview applications, as they need to offer an appropriate trade off between transmission rate or interaction flexibility and compression efficiency.

Therefore, we propose a greedy algorithm to find the optimal interview PS and QPs for the texture and depth maps. The optimal PS and QPs minimize the expected distortion at the user in a system where the point-to-point transmission bandwidth and the storage capacity are scarce resources. It is important to mention that the proposed algorithm is not specific to any coding standard, provided that we are using a temporal and interview predictive coding

solution for both texture and depth maps. We consider depth-image-based rendering (DIBR) techniques in order to render new views from encoded texture and depth maps. To better adapt the coding model to the video content along time, we characterize the user interaction behavior with a view popularity model [66] [67], assuming a random access interactivity model, where users can switch to any viewpoint in the multiview system and not only to neighboring views. Experimental results show that the proposed algorithm is able to identify a near-optimal PS in the sense of minimizing the distortion while trading off the transmission and storage costs. At the same time, our PS and associated QPs selection algorithm leads to a complexity reduction of up to 72% compared to an exhaustive search approach. Overall, the proposed algorithm permits to efficiently use the bandwidth available and storage capacity by optimizing inter-view dependencies on the PS, where the viewing user preferences are considered.

To achieve its objectives, this chapter is organized as follows: Section 3.2 outlines the main characteristics of the IMVS system under consideration. Section 3.3 describes the transmission and coding rate, and distortion models adopted in this work. Then, the optimization problem to find the optimal interview PS and associated QPs given some system constraints is formulated in Section 3.4. In Section 3.5, a greedy algorithm is proposed to efficiently solve the optimization problem, previously formulated. Section 3.6 presents and analyses the performance results demonstrating the benefits of the proposed solution considering both the MVC and 3D-HEVC coding standards and finally, the conclusions and further work are presented in Section 3.7.

3.2 IMVS Framework

We consider an IMVS system, where multiview video coding standards are used to compress texture and depth data for a limited set of views. The users may not only be interested in the coded viewpoints but also in intermediate viewpoints derived from a pair of textures and depths views. The most relevant characteristics of the IMVS system model considered in this work are described below.

3.2.1 Depth-based Multiview Model

We consider an IMVS system where a set of V views, $\mathcal{V} = \{1, \cdots, V\}$, is encoded at the sender side. For each coded view $v \in \mathcal{V}$, texture and depth maps are available, allowing the generation of intermediate virtual viewpoints at the decoder with an appropriate synthesis algorithm. This set of coded views may be different from the set of captured ones as the rate may be limited and/or the position of the cameras capturing the scene may not always be the optimal one. Between each pair of consecutive coded views, some virtual view positions may be available for user request at a minimum guaranteed quality. With the help of the depth-image based rendering (DIBR) technique, these views are synthesized using the closest right and left coded views, denoted as $\{v_R, v_L\} \in \mathcal{V}$. At the decoder side a view can be rendered at any

position in the discrete set $\mathscr{U} = \{1, 1+\delta, \cdots, V\}$; with δ as the minimum distance between consecutive views in the navigation window. The set of virtual views is defined as $\mathscr{W} = \mathscr{U} \setminus \mathscr{V}$.

Multiview video coding is applied to both texture and depth components of the set of coded views, using two coding standards: MVC+D and 3D-HEVC. Based on predefined storage and bandwidth constraints, both texture and depth images are encoded using the same optimized PS at their respective QPs, $Q = (q_t, q_d)$, where t and d stand for texture and depth, respectively. The QPs are typically different for the texture and depth data [68], as they have completely different impact on the final texture quality, and thus lead to different RD trade-offs. It is important to remember that, while decoded textures are directly offered to the users, decoded depths are not; they only serve to generate virtual views (thus also influencing their texture quality). All coded data is stored in a server and eventually transmitted when requested. The server provides an IMVS service to multiple users. We assume that when a coded view is requested by the user, only the texture information is transmitted. On the other hand, when a virtual view is requested, both the texture and depth maps of the closest right and left coded views are transmitted by the server, if not already available at the user, so that the user can synthesize the requested virtual viewpoint. This transmission model ensures the backward compatibility with traditional video decoders, by only offering texture information to users unable to synthesize virtual viewpoints. The same general IMVS system model can be considered in the stereo video case, where instead of one, two views are requested by the user, notably considering that different stereo displays may use different baseline distances. Then, if the two requested views are coded views, only their texture information is transmitted to the user. However, if one or both requested views are virtual views, texture and depth maps of the closest right and left coded views of one or both viewpoints need to be transmitted. Figure 3.1 illustrates the general IMVS system architecture, where the coded and virtual views are represented by frames connected by continuous and dashed arrows to the decoder and view synthesis blocks, respectively.

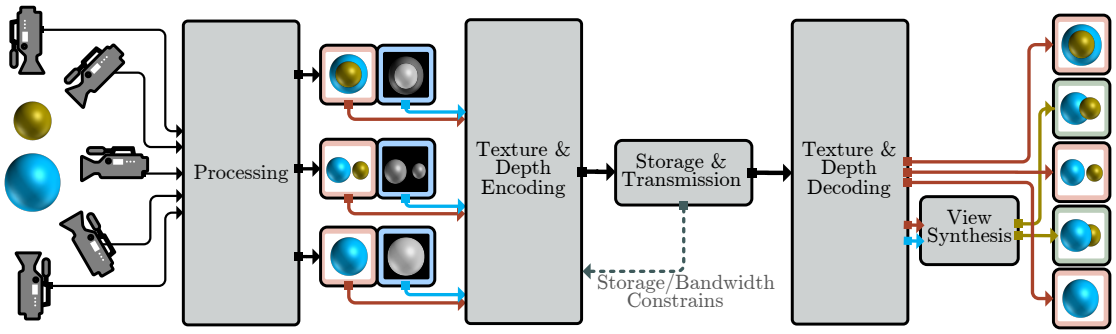


Figure 3.1 – General IMVS system architecture. Coded and virtual views are represented by images connected by continuous and dashed arrows to the texture decoder and view synthesis blocks, respectively.

3.2.2 Interactivity Model

In our system, we consider a *random access* interactivity model works and to model the user interaction, a view popularity factor, p_u^g, is considered to express the probability that a user selects view $u \in \mathscr{U}$ at the switching time instant (*i.e.*, at the anchor frames) of a group of pictures (GOP) g. We assume that the probability p_u^g, $\forall u \in \mathscr{U}$, depends on the popularity of the views or on the scene content itself but not on the view previously requested by the user. This may be the case for sports scenes for example, where a user may be following the moves of his/her favorite player but at a certain time decides to change to the most popular view, which is done independently of his/her current position. We assume a *static view temporal popularity model*, meaning that all the GOPs of a given view have the same probability of being requested by the users, although this may be easily modified if the temporal characteristics of the content are considered.

3.2.3 Coding Model

Multiview video plus depth coding considers both the temporal and interview correlations to increase the RD performance, reducing the redundancy among different views at the same time instance and among subsequent frames in time in the same view position. In this work, the same temporal and interview PS is used for coding both the texture and depth maps of the set of coded views $\mathscr{V}$. The temporal and interview coding models to be considered for the optimization of the texture and depth common PS have the following characteristics:

Temporal Coding Model

As commonly done in the literature, we assume a fixed temporal PS for each view (texture and depth maps), with hierarchical B-frames/slices [69], where B-frames are hierarchically predicted from other B or anchor frames. Figure 3.2 illustrates a typical hierarchical B-frames PS with 4 temporal layers, denoted with a sub-index from 0 to 3. The arrows in the figure indicate the reference frames used for the prediction of the various B frames. To control the quantization steps in the temporal domain, and thus the distortion, a cascading quantization parameters (CQP) [65] strategy is used. In this strategy, the full set of texture and depth QPs, $Q = (q_t, q_d)$, for the anchor frames are encoded with a small QP (high quality), since they are used as references for the prediction of frames in higher temporal layers. Then, the QPs of the frames in higher temporal layers are assigned by increasing the previous temporal layer QP with a pre-defined ΔQ, which may also be different for texture and depth. Here, we assume that even if q_t and q_d can take different values, their value distribution is the same for all the views in a particular GOP, as they vary at GOP level. Therefore, for a given PS there is only one q_t and one q_d that are used for all the texture and depth maps of the views, respectively. This assumption reduces the complexity of the Q search and it is not far from reality as the content of the various views from the same captured scene tends to be very similar and as a consequence the optimal QPs should also be similar among the views.

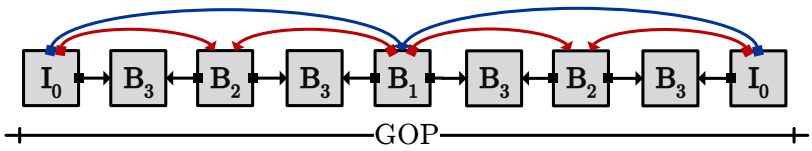


Figure 3.2 – Hierarchical B-frames with four temporal layers.

Interview Coding Model

The interview coding models considered here are based on the two most commonly used interview PSs in multiview video coding standards, namely IBP and IP [44]. In these PSs, hierarchical B-frames are used in the temporal domain while the IBP or IP modes are used for the anchor frames, determining the interview coding model of both anchor and non-anchor frames. Although the use of interview coding in the non-anchor frames is optional, here we use it as it has been shown to improve the RD performance in typical sequences [44]. Typically, in IBP and IP PSs only one independently encoded view or *key view* is considered (*e.g.*, a lateral view) in order to maximize the interview redundancy reduction. However, as high compression efficiency is not the only objective in IMVS systems, we allow here more than one key view in the two basic PSs (IBP and IP) to reduce the coding dependencies and increase the navigation flexibility. To illustrate this, let us consider the example in Fig. 3.3 where two IP - PSs are shown, one with only one key view (view 0) and the other with two key views (views 0 and 2). We further show (in gray) the frames that need to be transmitted in order to decode a GOP in view 3. It can be seen that, due to the interview coding dependencies, for the PS with only one key view (maximum compression efficiency) all the frames from the previous views (views 0, 1 and 2) need to be transmitted together with the requested view 3, while for the PS with two key views, only views 2 and 3 need to be transmitted. Finally, for benchmarking purposes, we also consider the simulcasting structure where all the views are key views (I-PS).

3.3 Rate and Distortion Modeling

To fully characterize the IMVS system, we now define the rate and distortion models considered in this work. In the following, we use F to denote the frame texture and $\mathscr{F}$ to denote both the texture and the depth components of a frame. Both F and $\mathscr{F}$ refer to frames fully covered by a single type of slice, namely I-, P- or B-slices.

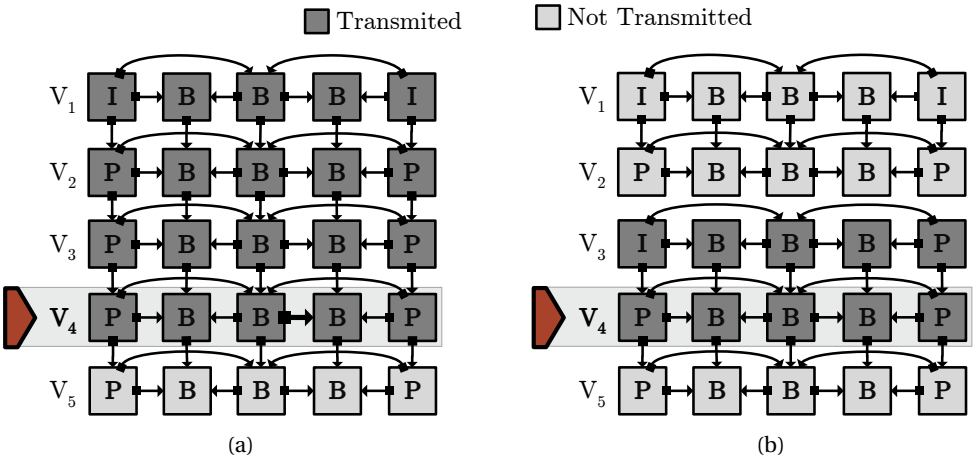


Figure 3.3 – Interview coding dependencies example. Two IP PSs are illustrated along with the coding dependencies: (a) with only one key view (view 1) and (b) with two key views (views 1 and 3). The frames that need to be transmitted, in order to decode a GOP from view 4 are shown in grey.

3.3.1 Coding Rate

The coding rate (*CR*) is defined as the total number of bits per unit of time necessary to code both the texture and depth maps of a multiview sequence and it may be computed as:

$$CR = f\frac{\sum_{g=1}^{G}\sum_{v=1}^{V}\sum_{n=1}^{N} n_b\left(\mathscr{F}_{v,n}^{g}\left(PS_g, Q_g\right)\right)}{GNV} \tag{3.1}$$

where f is the frame rate in frames per second, G the total number of GOPs per view, V the number of coded views, N the number of frames per view in a GOP (we assume that all the views have the same GOP size) and $n_b\left(\mathscr{F}_{v,n}^{g}\right)$ the number of bits used to code frame $\mathscr{F}_{v,n}^{g}$ of view $c \in \mathscr{V}$ at time instant n in GOP g. The number of bits necessary to code frame $\mathscr{F}_{v,n}^{g}$ depends on the PS and the set of QPs used to code the texture and depth on each particular GOP g, PS_g and $Q_g = \left(q_t^g, q_d^g\right)$. It is important to mention that since we consider that the PS may vary on a GOP basis, also the texture and depth QPs should vary in order to better match the system constraints. Typically, a PS with only one key view, meaning a maximum number of interview dependencies, and a coarse quantization should result in higher compression efficiency or lower coding rate, *CR*, in Eq. (3.1).

3.3.2 Transmission Rate

The transmission rate (TR) is here associated to a point-to-point connection where a dedicated video stream is transmitted between two network nodes. This transmission model is useful

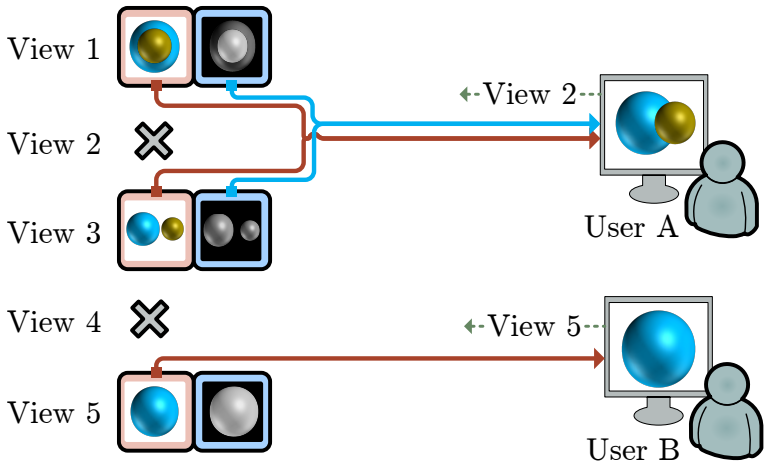


Figure 3.4 – Transmission model example where views $\{1,2,3,4,5\} \in \mathcal{U}$; $\{1,3,5\} \in \mathcal{V}$ and $\{2,4\} \in \mathcal{W}$. User A requests virtual view 2 and User B coded view 5 (dashed arrows). Coded views 1 and 3 have to be transmitted to user A, in order to synthesize the requested virtual view, while for user B only the texture information of view 5 need to be sent.

in content on-demand scenarios where users act independently; hence there are not many streams that could be shared between them as normally the probability that two or more users request the same video stream at the same time is very low. The *TR* depends on the PS considered, in particular on the interview PS. For instance, in order to decode a particular frame, other frames from the same time instant but from different views might have to be transmitted and processed before decoding the requested view. This is illustrated in Fig. 3.3, where an example of the effect of interview dependencies is presented. In addition, the *TR* also depends on which view is requested by the user, notably whether it is a coded or virtual view. If the requested view is a coded view, $v \in \mathcal{V}$, only its texture information has to be transmitted. Otherwise, if the requested view is a virtual view, $w \in \mathcal{W}$, both the texture and depth maps of the closest right and left coded views have to be transmitted, if not already available, so that the user can synthesize the requested virtual viewpoint. This is illustrated in Fig 3.4, where user A requests a virtual view (view 2) while user B asks for a coded view (view 5). Then, coded views 1 and 3 (texture and depth maps) have to be transmitted to user A, in order to synthesize the requested virtual view, while for user B, only the texture information of view 5 has to be sent.

Before defining the transmission rate TR, we need to define the so-called frame- and GOP-dependency path size. Similar to the transmission cost defined in [35], the frame-dependency path size $\phi\left(F^g_{u,n}\right)$ corresponds to the number of bits that have to be transmitted to be able to decode or synthesize a particular texture frame from view $u \in \mathcal{U}$. The definition of $\phi\left(F^g_{u,n}\right)$ depends on whether $F^g_{u,n}$ corresponds to a frame in a coded view, $u = v | v \in \mathcal{V}$, or from a virtual view, $u = w | w \in \mathcal{W}$. If $F^g_{u,n}$ corresponds to a frame in a coded view, $F^g_{u,n} = F^g_{v,n}$, $\phi\left(F^g_{v,n}\right)$ is

recursively defined as:

$$\phi\left(F^g_{v,n}\right) = n_b\left(F^g_{v,n}\left(PS_g, q^g_t\right)\right) + \sum_{\hat{v}\in\{c-1,c+1\}} \phi\left(F^g_{\hat{v},n}\right) + \sum_{\hat{n}\in\{1,\cdots,N\}\setminus n} \phi\left(F^g_{v,\hat{n}}\right) \tag{3.2}$$

where $F^g_{\hat{v},n}$ and $F^g_{v,\hat{n}}$ are the spatial and temporal reference frames for $F^g_{v,n}$, respectively. The frame $F^g_{\hat{v},n}$ corresponds to the reference frame of $F^g_{v,n}$ from the same time instant but from one of the two neighboring views (depending on the interview PS), while frame $F^g_{v,\hat{n}}$ is a reference frame from the same view $v \in \mathcal{V}$ and GOP g, but at different time instant. In (3.2) each frame is considered once, so redundancy is avoided.

On the other hand, if $F^g_{u,n}$ corresponds to a frame from a virtual view, $F^g_{u,n} = F^g_{w,n}$, the texture and depth data of the closest right and a left coded view, $\mathscr{F}^g_{v_R,n}$ and $\mathscr{F}^g_{v_L,n}$, for $\{v_R, v_L\} \in \mathcal{V}$, need to be transmitted and decoded in order to synthesize frame $F^g_{w,n}$. Therefore, $\phi\left(F^g_{w,n}\right)$ becomes:

$$\phi\left(F^g_{w,n}\right) = \phi\left(\mathscr{F}^g_{v_R,n}\right) + \phi\left(\mathscr{F}^g_{v_L,n}\right) \tag{3.3}$$

where, $\phi\left(\mathscr{F}^g_{v_R,n}\right)$ and $\phi\left(\mathscr{F}^g_{v_L,n}\right)$ are still the frame dependency paths of frames $\mathscr{F}^g_{v_R,n}$ and $\mathscr{F}^g_{v_L,n}$, where both texture and depth data are considered. Remember that here we consider that texture and depth data use the same optimized PS, so that the coding dependencies are the same for both data types. Then, $\phi\left(\mathscr{F}^g_{v_R,n}\right)$ and $\phi\left(\mathscr{F}^g_{v_L,n}\right)$ are recursively defined as in (3.2); for instance, in the case of $\phi\left(\mathscr{F}^g_{v_R,n}\right)$ we have:

$$\phi\left(\mathscr{F}^g_{v_R,n}\right) = n_b\left(\mathscr{F}^g_{v_R,n}\left(PS_g, Q_g\right)\right) + \sum_{\hat{c}\in\{c-1,c+1\}} \phi\left(\mathscr{F}^g_{\hat{c},n}\right) + \sum_{\hat{n}\in\{1,\cdots,N\}\setminus n} \phi\left(\mathscr{F}^g_{c,\hat{n}}\right) \tag{3.4}$$

The frame-dependency path size for the left reference view, $\phi\left(\mathscr{F}^g_{v_L,n}\right)$, is similarly defined.

As a consequence, the number of bits required to decode or synthesize all the frames in a GOP g of a particular view $u \in \mathcal{U}$, named GOP-dependency path size ϕ^g_u, is defined as:

$$\phi^g_u = \sum_{n=1}^{N} \phi\left(F^g_{u,n}\right) = \begin{cases} \sum_{n=1}^{N} \phi\left(F^g_{v,n}\right), & \text{if } u = v | v \in \mathcal{V} \\ \sum_{n=1}^{N} \phi\left(F^g_{w,n}\right), & \text{if } u = w | w \in \mathcal{V} \end{cases} \tag{3.5}$$

where each frame $F^g_{u,n}$ is considered only once. We assume that $F^g_{u,n}$ stays at the decoder side

for at least the duration of the current GOP g, so it does not need to be re-transmitted if it is required for decoding a future frame. We also assume that frame dependencies are limited to a GOP.

Finally, we compute the overall *expected point-to-point transmission rate, TR*, as:

$$TR = f\frac{\sum_{g=1}^{G} E\{\phi_u^g\}}{GN} \tag{3.6}$$

where $E\{\phi_u^g\}$ is the expectation of the GOP-dependency path size ϕ_u^g, which is defined as $E\{\phi_u^g\} = \sum_{u=1}^{U} p_u^g \phi_u^g$, considering the view popularity model, p_u^g, to express the user preferences for the various views in a particular GOP, common for all the views. Differently of the *CR*, assuming that the texture and depth QPs are fixed, by increasing the GOP-dependency path size (*i.e.*, increasing the number of interview dependencies), the *TR* increases, as more frames need to be transmitted in order to decode or render a particular frame.

3.3.3 Distortion

The average distortion for GOP g in view u, D_u^g, corresponding to the coding noise associated to the quantization process, is taken as the temporal average of the distortion per frame in GOP g, $D_{u,n}^g$:

$$D_u^g = \frac{\sum_{n=1}^{N} D_{u,n}^g}{N} \tag{3.7}$$

If the view $u \in \mathcal{U}$ corresponds to a coded view, $u = v | v \in \mathcal{V}$, its distortion D_v^g depends only on the texture QP, q_t^g. Otherwise, if u is a virtual view, $u = w | w \in \mathcal{V}$, its distortion D_w^g, depends on both the texture and the depth QPs, $Q_g = (q_t^g, q_d^g)$, used to encode the right and left reference views; $\{v_R, v_L\} \in \mathcal{V}$. The distortion perceived by the user for a particular GOP g takes the value D_u^g with probability p_u^g (*i.e.*, the view popularity factor). Then, the expected distortion in a specific GOP g, D_g, for the multiview sequence is defined as:

$$D_g = \sum_{u=1}^{U} p_u^g D_u^g = \sum_{v=1}^{V} p_v^g D_v^g(q_t^g) + \sum_{w=1}^{W} p_w^g D_w^g(Q_g) \tag{3.8}$$

Note that the distortion of both coded and virtual views, D_v^g and D_w^g, mainly depends on the QPs of the coded or reference views and not on the PS chosen.

We measure the distortion due to different coding choices in order to select the best coding strategy. To quantify the distortion of the coded views, D_c^g, we measure the mean-squared-error (MSE) between the original view and its coded version. Regarding the distortion of the virtual views, D_v^g, typically there are no original frames available to compute the same metric or any full reference objective quality metric. A commonly used solution available in the literature, and adopted in this chapter, consists in computing a virtual reference view from the uncompressed texture and depth data of the closest right and left coded views. Then, this synthetic view is taken as benchmark to evaluate the distortion, *e.g.* the MSE, of the same view synthesized from the decoded reference views [70]. Alternatively, one could use a distortion model for the virtual views, instead of computing it explicitly using the available data (Chapter 4).

Finally, the expected distortion for the overall multiview sequence is defined as:

$$D = \frac{\sum_{g=1}^{G} D_g}{G} \tag{3.9}$$

In this chapter, for the sake of simplicity, we use the terms distortion and transmission rate when referring to the expected distortion and expected point-to-point transmission rate per sequence, respectively.

3.4 Problem Formulation

After describing the main characteristics of our IMVS system, we shall now formulate the optimization problem. The problem addressed here is to find the optimal texture and depth interview PS per GOP, $PS^* = \{PS_1^*, PS_2^*, \cdots, PS_G^*\}$, together with their associated optimal texture and depth QPs, $Q^* = \{Q_1^*, Q_2^*, \cdots, Q_G^*\}$ to encode a predefined set of views, minimizing the distortion D while considering the following storage and bandwidth related constraints:

- *Storage constraint* — For convenience, we express the storage capacity of the system as a rate, *CR*, notably as the total number of bits per unit of time used to code all the views, considering both texture and depth. The constraint states that the coding rate shall not exceed the maximum storage capacity of the system, CR_{max}.

- *Bandwidth constraint* — Moreover, the transmission rate, TR, for each user is limited by the maximum data rate supported by the network for any user, namely TR_{max}.

In summary, the optimization problem may be written as follows:

$$\{PS^*, Q^*\} = \underset{PS,Q}{\operatorname{argmin}} \; D(Q) \tag{3.10}$$

such that,

$$CR(PS,Q) \leq CR_{max} \qquad \text{Storage constraint}$$
$$TR(PS,Q) \leq TR_{max} \qquad \text{Bandwidth constraint}$$

where *CR*, *TR* and *D* are calculated as in (3.1), (3.6) and (3.9), respectively. When all the GOPs have the same probability of being requested by the user, meaning a static view temporal popularity model is assumed, the optimization problem defined in (3.10) can be independently solved for each GOP. Then, the optimal PS per GOP g, PS_g^* and associated texture and depth QPs, Q_g^*, corresponds to those minimizing the GOP distortion, D_g, as defined in (3.8):

$$\left\{PS_g^*, Q_g^*\right\} = \underset{PS_g,Q_g}{\operatorname{argmin}} \; D_g\left(Q_g\right) \tag{3.11}$$

such that,

$$CR_g = \frac{f}{NC}\sum_{c=1}^{C}\sum_{n=1}^{N} n_b\left(\mathscr{F}_{c,n}^{g}\left(PS_g, Q_g\right)\right) \leq CR_{max}$$
$$TR_g = \frac{f}{N}E\left\{\phi_u^g\right\} \leq TR_{max}$$

where the expressions for the storage and bandwidth constraints are calculated from (3.1) and (3.6), respectively.

For the sake of simplicity, we assume that an optimal bitrate allocation (eventually at GOP level) between texture and depth is known. A different texture and depth rate ratio is expected for different sequences, as it has been shown to be content dependent [71] [72].

$$CR_d^g \leq CR_{d,max} \qquad CR_t^g \leq CR_{t,max} \tag{3.12}$$

Solving the combinatorial optimization problem defined in (3.11) can be very computationally intensive, notably if exhaustive search (ES) is applied. Indeed, the number of possible interview

PSs exponentially grows with the number of views in the multiview set, and for each PS multiple texture and depth QPs configurations are possible. Therefore, in the following section we propose a greedy algorithm that finds near-optimal PSs and associated texture and depth QPs, with remarkably reduced complexity, able to minimize the distortion under storage and bandwidth constraints.

3.5 Optimization Algorithm

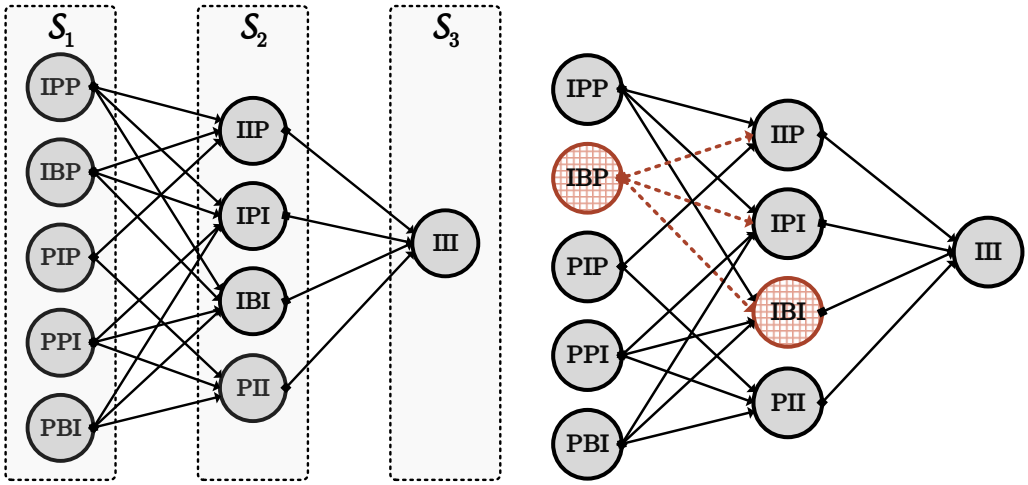
In this section, we propose a novel optimization algorithm that is able to find, for each GOP over all views, with a reduced complexity, a near-optimal PS with associated texture and depth QPs, given some IMVS system constraints. To significantly reduce the overall complexity regarding an exhaustive search (ES) approach, we propose a greedy optimization solution, which basically reduces the set of considered PSs without significant compression performance penalty. With this approach, the problem in (3.11) is solved by breaking it down into a series of stages, S_i, which are successively solved, one after the other. To better understand these different stages and how they depend on each other, we adopt a graph to embody all this information. Then, for each GOP over all views, the optimization problem in (3.11) is solved based on this stage graph.

3.5.1 Stage Graph Creation

The stage graph defines the various phases of the solution for the problem in Eq. (3.11). Each stage S_i includes a set of associated states representing the possible PS solutions at each phase of the proposed algorithm. These PSs are then processed in order to find the best PS and QPs in the stage, denoted as PS_i^{g*} and Q_i^{g*}. The states of consecutive stages are linked if they contain a similar sub-structure, which is defined in terms of key views position. In the following, we describe the two main steps in the stages graph creation process, notably the states and links definition.

States Definition

We define the states in our stages graph in terms of the number of key views in the interview prediction structure. Thus, the states in a particular stage correspond to the PSs with the same number of key views (*e.g.*, $1,2,\cdots,V$), in different positions of the multiview set. We start by including in the first stage, S_1, all possible PSs (for the IBP and IP PSs considered in this paper) with only one key view. This corresponds to the solutions with the maximum number of interview coding dependencies, thus associated with maximum compression efficiency and also maximum transmission rate in an IMVS system. Then, we gradually increase the number of key views in the PSs as we move towards the following stages, until the last stage, S_V, where all the V views are independently encoded. This corresponds to the absence of interview



(a) Stages states and link definition

(b) Sub-stage SS_2 = {IIP, IPI, IBI}, given that PS_1^{g*} = IBP.

Figure 3.5 – Example of a three stages graph definition and PS selection.

coding dependencies, hence minimum compression efficiency and minimum transmission rate in an IMVS system. Therefore, for fixed texture and depth QPs, by moving from stage S_1 to stage S_V, we are, in general, moving along solutions from a maximum *TR* (minimum *CR*) to a minimum *TR* (maximum *CR*), as the redundancy between the views increases *i.e.*, the number of interview dependencies decreases.

Links Definition

To link the states of two consecutive stages, we assume that the optimal PS, PS_i^{g*}, in a particular stage S_i for a specific GOP g, determines the optimal position of the i key views in the final optimal PS. This means, for example, that the optimal PS in S_1 determines the positioning of one of the key views in the optimal PS solution. Therefore, a link is defined between two states j, k, associated to $PS_{i-1,j}^g$ and $PS_{i,k}^g$ from stages S_{i-1} and S_i, if the $i-1$ key views in $PS_{i-1,j}^g$ keep their position in $PS_{i,k}^g$. This is illustrated in Fig. 3.5a where the different states are represented by circles and the links are defined between PSs of consecutive stages that preserve the key views position. The set of PSs in stage S_i linked to a same PS in stage S_{i-1}, $PS_{i-1,j}^g$, is called a sub-stage of S_i and denoted as $SS_{i,j}$, given $PS_{i-1,j}^g \in S_{i-1}$. In this work, there is only one sub-stage relevant for each stage, this means the one corresponding to the optimal PS in the previous stage. Therefore, to shorten the sub-stage notation, here a sub-stage of S_i is denoted as SS_i, which is associated to PS_{i-1}^{g*}, while $|SS_i|$ stands for the number of states in SS_i. For instance, in Fig. 3.5b, the IIP, IPI and IBI PSs define SS_2, given that PS_{i-1}^{g*} = IBP. In the particular case of stage S_1, $SS_1 = S_1$, as there is no previous stage.

3.5.2 Iterative PS and Texture and Depth QPs Selection

The stages of the graph are successively processed for each GOP of the multiview sequence, starting with stage S_1, until the adopted stopping criterion is fulfilled, meaning that the best PS for a particular GOP g, PS_g^*, (defined over all the coded views) has been found together with the optimal texture and depth QPs, Q_g^*. At each stage S_i, only the PSs in the sub-stage SS_i, given PS_{i-1}^{g*}, are processed to find the optimal PS and Q, this means PS_i^{g*} and Q_i^{g*}.

The optimal PS for GOP g and sub-stage SS_i, PS_i^{g*}, and associated optimal texture and depth QPs, Q_i^{g*}, are found by alternatively solving the problem in (3.11) for the PSs and QPs in SS_i. In particular, the following steps are followed for each S_i, starting with S_1:

Initialize $Q_{i,j}^g$

For each PS in sub-stage SS_i, find $Q = (q_t, q_d)$ that satisfies the texture and depth components of the storage constraint, as defined in (3.12). We denote it as $Q_{i,j}^g$, which is associated to $PS_{i,j}^g$ from sub-stage SS_i and state $j \in SS_i$, as Q may take different values for different PSs in SS_i. By initializing $Q_{i,j}^g$, for each PS in SS_i, $PS_{i,j}^g$, such that it satisfies one of the problem constraints in (3.11), we are trying to find a set of texture and depth QPs that is close enough to the optimal one. Here, we have only considered the storage constraint, but the bandwidth constraint could have been also used if preferred.

Find Optimal PS, PS_i^{g*}

Here, we optimize the problem in (3.11) only for the PSs in SS_i, while the texture and depth QPs set is kept fixed for each PS. In particular, we consider $Q_{i,j}^g$ as the QPs set for each PS in SS_i. Hence, the problem addressed here is to find the optimal PS in SS_i for the GOP g, PS_i^{g*}, that minimizes the GOP distortion D_g given some storage and bandwidth constraints:

$$PS_i^{g*} = \underset{PS_{i,j}^g}{\operatorname{argmin}} \; D_g\left(PS_{i,j}^g\right), \quad \forall PS_{i,j}^g \in SS_i \tag{3.13}$$

such that,

$$CR_g \leq CR_{max} \qquad\qquad TR_g \leq TR_{max}$$

where we do not consider the texture and depth components of the CR independently, as for each PS we have already found the texture and depth QPs fulfilling the storage constraint for the texture and depth maps (Section 3.5.2).

To solve the combinatorial problem in (3.13), we apply the Lagrangian relaxation approach, where according to [73] the constraints are first relaxed by adding them into the objective function with an associated weight (the Lagrangian multiplier). In our case, we move the storage and bandwidth constraints, as in (3.13), to the objective function with the Lagrangian multipliers, $\{\lambda, \mu\} \geq 0$. Each Lagrangian multiplier represents a penalty to be added to a solution that does not satisfy the considered constraints. Then, the problem in (3.13) is relaxed as follows:

$$\mathscr{J}_i\left(PS_{i,j}^g, \lambda, \mu\right) = \min_{PS_{i,j}^g} \left\{D_g - \lambda\left(CR_{max} - CR_g\right) - \mu\left(TR_{max} - TR_g\right)\right\} \tag{3.14}$$

In (3.14), we have eliminated the constraints from (3.13), but the number of variables has increased with the number of eliminated constraints or the number of Lagrangian multipliers used. To find the optimal values for the Lagrangian multipliers, λ and μ, we solve the Lagrangian dual problem [73]:

$$\{\lambda^*, \mu^*\} = \underset{\lambda,\mu}{\operatorname{argmax}} \ \mathscr{J}_i \tag{3.15}$$

Finally, considering only the PSs in SS_i, the best PS for GOP g and stage S_i, PS_i^{g*}, is the one minimizing (3.14) for the optimal Lagrangian multipliers obtained in (3.15).

Find Optimal Q, Q_i^{g*}

Given the optimal PS in SS_i and GOP g, PS_i^{g*}, the problem addressed here is to find the optimal set of texture and depth QPs, Q_i^{g*} minimizing the distortion given some storage and bandwidth constraint:

$$Q_i^{g*} = \underset{Q}{\operatorname{argmin}} \ D_g(Q) \tag{3.16}$$

such that,

$$CR_d^g \leq CR_{d,max} \qquad CR_t^g \leq CR_{t,max} \qquad TR_g \leq TR_{max}$$

Differently from the problem posed in (3.13), here we consider the texture and depth components of the CR independently, as we need to find the set of texture and depth QPs, Q_i^{g*},

satisfying these constraints, while in (3.13), for each PS, we have already selected the set Q satisfying both of the *CR* constraints.

As in Section 3.5.2, to solve the problem in (3.16), we apply the Lagrangian relaxation approach with the Lagrangian multipliers, $\{\alpha, \beta, \gamma\} \geq 0$:

$$\mathscr{L}_i\left(Q, \alpha, \beta, \gamma\right) = \min_{Q} \left\{D_g - \alpha\left(CR_{t,max} - CR_t^g\right) - \beta\left(CR_{d,max} - CR_d^g\right) - \gamma\left(TR_{max} - TR_g\right)\right\} \tag{3.17}$$

In order to find the optimal α, β and γ values, we solve the following Lagrangian dual problem:

$$\{\alpha^*, \beta^*, \gamma^*\} = \underset{\alpha,\beta,\gamma}{\operatorname{argmax}}\ \mathscr{L}_i \tag{3.18}$$

Then, the best Q for GOP g and stage S_i, Q_i^{g*}, is the one minimizing (3.17) for the optimal Lagrangian multipliers obtained in (3.18).

The optimal PS, PS_i^{g*}, found in Section 3.5.2 using $Q_{i,j}^g$, with high probability, is not changed after modifying the texture and depth QPs to the optimal ones, Q_i^{g*}. This is due to the similarities between PSs compared in each stage of our algorithm. This statement is further justified in Section 3.5.4. Therefore, there is not need to recalculate the optimal PS of the current sub-stage for the new texture and depth QPs, Q_i^{g*}.

Before moving to the following stage, the stopping criterion needs to be checked. This is explained in the following.

3.5.3 Stopping Criterion Checking

The decision to process the next stage in the graph or to stop the PS selection algorithm at the current stage depends on the fulfillment of the following stopping criterion. If $\mathscr{L}_i$ is larger than $\mathscr{L}_{i-1}$, then stop the optimization algorithm as $PS_g^* = PS_{i-1}^{g*}$ and $Q_g^* = Q_{i-1}^{g*}$ define the locally optimum solution, since moving to the next stage will increase the Lagrangian cost, which is not desirable. In other words, by moving from stage S_{i-1} to stage S_i, at a fixed quality, we are in general moving to solutions with higher coding rate and lower transmission rate, as the number of interview coding dependencies decreases from one stage to the other. Then, an increase of $\mathscr{L}_i$, as defined in (3.17), means that the distortion has increased in order to satisfy the storage capacity constraint. Thus, as we move forward to the following stages, after the first increase of the $\mathscr{L}_i$ value, we expect $\mathscr{L}_i$ to monotonically increase, as the *CR* will become higher (for a fixed quality level). As a result, moving to upcoming stages, after the rise of the

Lagrangian cost $\mathscr{L}_i$, will only increase the complexity of the algorithm with no benefits in terms of reduced distortion.

It is important to mention that, if S_i is the last stage of the graph, $S_i = S_V$, and $\mathscr{L}_i < \mathscr{L}_{i-1}$ then the locally optimal solution is defined by the current solution $PS_g^* = PS_i^{g*}$ and $Q_g^* = Q_i^{g*}$.

Following this approach we achieve a major reduction on the complexity associated to solving the optimization problem at the price of slightly losing optimality. Although this greedy algorithm determines the optimal PS (and associated optimal Q) at a sub-stage level, the final PS may not be the global optimal one, as at each stage some PSs are ignored. Remind that under the assumptions made, there is only one sub-stage relevant for each stage, this means the one corresponding to the optimal PS in the previous stage. However, a good performance is expected as when adding a new key view at each stage, it is very unlikely that the previous k views do not maintain their optimal position in the multiview set. This argument becomes stronger as k becomes larger as the key views positions providing higher gain are chosen in the first stages of the algorithm. This is confirmed with the experimental results.

The flowchart in Fig. 3.6 summarizes our optimization algorithm, after the creation of the stage graph.

3.5.4 Sub-stage Optimal PS and Q Relationship

We discuss here why it is reasonable to claim that, at each stage of our greedy algorithm, the optimal PS, PS_i^{g*}, tends to be independent of the level of quality or the QPs used to encode the texture and depth maps, $Q = (q_t, q_d)$. This is important to justify the decision taken in Section 3.5.2 of not recalculating the optimal PS for the obtained Q_i^{g*}.

For the PSs considered in this work, the various views are different in terms of the type of coding used at the anchor frame time, meaning an I- P- or B-frame (meaning frames with I, P or B slices). Empirically, we have seen that for the same coding conditions, each type of frame, I, P (in anchor frame position) and B (in anchor and non-anchor frame position) tends to have the same number of bits as another frame of the same coding type in a different view but at the same time instant. This is true because, we compare frames with similar motion characteristics, as they are frames from the same time instant and the scene is typically captured with equidistant cameras. Then, when we compare the *CR* between the PSs with the same quality and the same number of key views, as it is done at each stage of the graph of our greedy algorithm, the number of bits required for each PS is very similar. In general, the *CR* values are closer for IP PSs than for IBP PSs, as IP PSs are more similar than IBP PSs. In particular, for the same number of key views, IP PSs have the same number of P anchor frames while IBP PSs may have different number of B and P frames, depending on the position of the key views. On the other hand, when, for a particular quality level, we compare the *TR* between the PSs with the same number of key views (from the same graph stage), the expected number of bits per unit of time that is needed to decode a view may change from one PS to another.

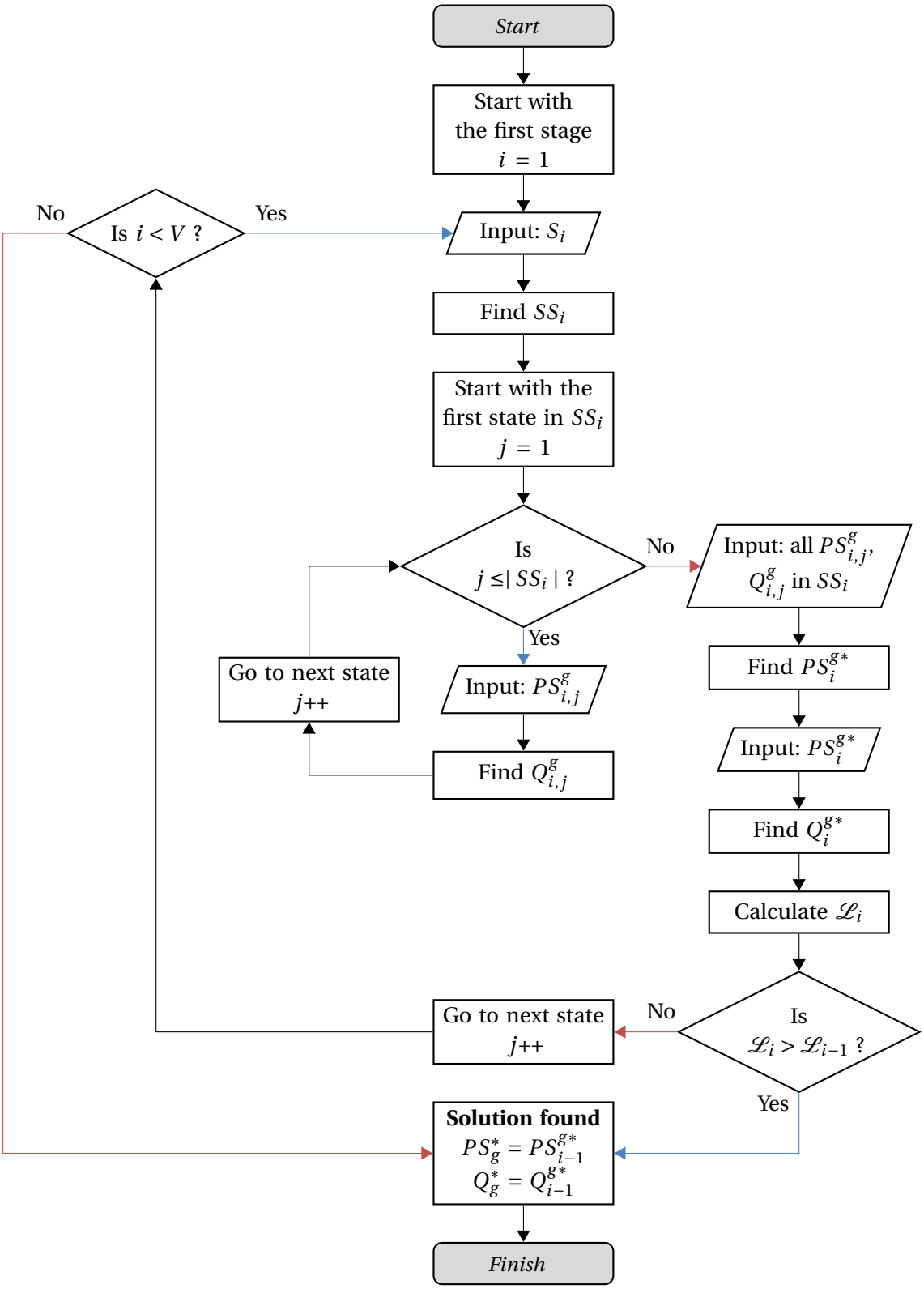


Figure 3.6 – Flowchart of the proposed optimization algorithm, after the stage graph creation.

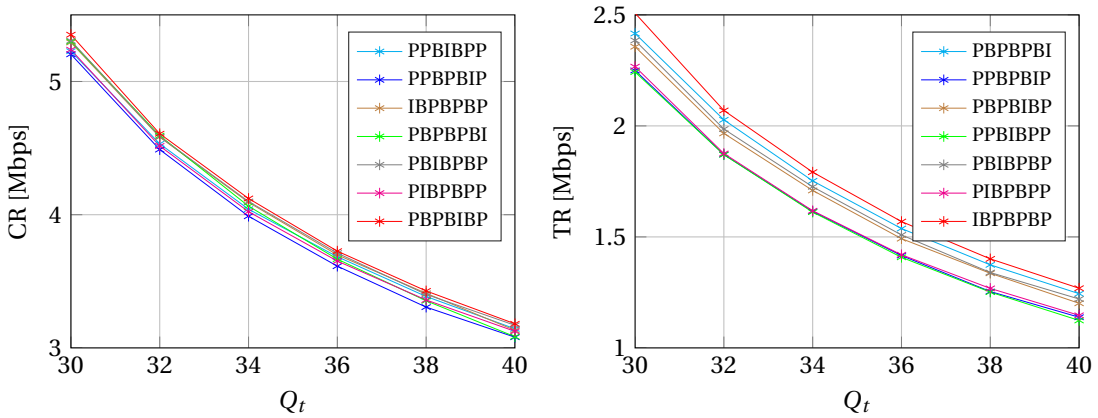


Figure 3.7 – Relationship between (a) *CR* and Q_t, and (b) *TR* and Q_t for IBP PSs with one key view. *Poznan_Hall2* [3] sequence is considered, where a total of $U = 13$ views are available for request ($C = 7$ and $V = 6$).

This occurs since the relative position of the different views in the multiview set and the view popularity distribution have a great impact on the *TR* value (please, refer to Section 3.3.2). However, as explained before, the PSs compared have, most of the time, the same frame types, as they are PSs from the same graph stage and almost the same number of bits for each frame type. Therefore, as the QP decreases (increases), we expect that the proportion of the increase (decrease) of the transmission rate is the same for all the PSs in the same graph stage. This means that the *TR* difference between PSs at different QPs is very much constant, making the optimal PS independent of the QP selected.

This can be better understood through an example. Let us consider the multiview sequence *Poznan_Hall2* [3] where $V = 7$ coded views and one virtual view between each pair of coded views are considered, for a total of 13 views. Let us also assume a uniform popularity distribution, which means $p_u^g = 1/13$, $\forall u \in \mathscr{U}$. The seven views available at the server side are encoded using the MVC reference software JMVC v8.2 [74] with all the possible IBP PSs with one key view (corresponding to the IBP PSs in the first stage of our greedy algorithm), where the texture QP, q_t varies and the depth QP is kept fixed, $q_d = 42$. Figures 3.7a and 3.7b show the relationship between *CR* and q_t and *TR* and q_t, respectively, for all IBP PSs with a single key view this means in stage S_1. The charts show that for different IBP PSs the *CR* is very similar, where the number of P and B views may be different as they depend on the position of the key views. For *TR*, PSs with less B-frames as anchor frames tend to have better performance. However, the curves representing the efficiency of the PSs (in terms of *CR* or *TR*) are rather parallel, for both *CR* and *TR*, which means that the efficiency difference between the PSs is independent of the quality level. Therefore, the optimal PS in a particular stage of our greedy algorithm is very much independent of the quality level. The same behavior has been observed for IP PSs and for PSs with more than one key view.

3.6 Performance Assessment

This section presents the test conditions and performance results obtained in different scenarios when the PS and associated texture and depth maps QPs search is performed with our proposed algorithm.

3.6.1 Content and Coding Test Conditions

As multiview video coding standards, we have considered the MVC, with the reference software JMVC v8.2 [74], and the 3D-HEVC, with the reference software HTM 6.2 [75]. As multiview data, we have used the sequences *Poznan_Hall2* [3] (1920 × 1080, 25Hz), *Pantomime* [76] (1280 × 960, 30Hz), *Book Arrival* [77] (1024 × 768, 16.67Hz), *GT_Fly* [78] (1920 × 1080, 25Hz) and *Undo Dancer* [79] (1920 × 1080, 25Hz). Figure 3.8 illustrates some frames of the considered sequences. While *Poznan_Hall2*, *Pantomime* and *Book Arrival* are real captured scenes, *GT_Fly* and *Undo Dancer* are computer-generated scenes. For all sequences, a GOP size of 8 frames has been adopted as specified in JCT-3V common test conditions [80]. In the temporal domain, the CQP strategy has been used with a fixed ΔQ equal to 0, 3 and 1 when the temporal layer was equal to 0, 1 and larger than 1, respectively. This is a common ΔQ setting for multiview test sequences. For each sequence, the following conditions have been considered:

- *Poznan_Hall2* [3] — $|\mathcal{V}| = 7$ coded views and $|\mathcal{W}| = 6$ virtual views, each located between two coded views. The seven coded views correspond to the views captured by the first seven cameras. The cameras are horizontally arranged with a fixed distance between neighboring cameras of approximately 13.75 cm.

- *Pantomime* [76] — $|\mathcal{V}| = 10$ coded views and $|\mathcal{W}| = 9$ virtual views, each located between two coded views. The ten coded views correspond to the captured views $\mathcal{V} = \{34-43\}$. The cameras are horizontally arranged with a fixed stereo distance.

- *Book Arrival* [77] — $|\mathcal{V}| = 5$ coded views and $|\mathcal{W}| = 4$ virtual views, each located between two coded views. The five coded views correspond to the captured views $\mathcal{V} = \{6,7,8,9,10\}$. The cameras are horizontally arranged with a spacing of 6.5 cm.

- *GT_Fly* [78] — The five available views are taken as coded views, $\mathcal{V} = \{1,2,3,5,9\}$, and we consider four virtual views $\mathcal{W} = \{4,6,7,8\}$. In this sequence, cameras are equidistantly arranged but the camera separation changes with time in order to preserve the 3D perception of the various scenes types: “landscape-view” and “near-view” scenes.

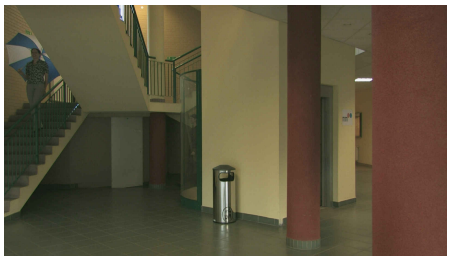
- *Undo Dancer* [79] — As for *GT_Fly*, the five available views are taken as coded views, $\mathcal{V} = \{1,2,3,5,9\}$, and we consider four virtual views $\mathcal{W} = \{4,6,7,8\}$. The cameras for this sequence are horizontally arranged with a fixed distance of 20 cm between neighboring views; this means that there are 80 cm of separation between the captured views 5 and 9.

For the sequences *Poznan_Hall2*, *Pantomime* and *Book Arrival* not all the depth maps for the coded views are provided. Therefore, we used the MPEG depth estimation reference software (DERS) [81] to generate the missing depth maps of these three sequences. In addition, we used the MPEG view synthesis reference software (VSRS) [82] based on DIBR, to synthesize the virtual views of all the considered sequences.

Depending on the content characteristics, this means after visual inspection, we have assigned different view popularity distributions to different sets of frames in the considered sequences. The view popularity distributions assumed here are: uniform (equally distributed popularity among the views), exponential (most popular views are located at the left end of the multiview set), inverted exponential (most popular views are located at the right end of the multiview set), Gaussian (most popular views are located at the center of the multiview set) and U-quadratic (most popular views are located at the borders of the multiview set). Table 3.1 shows the frame sets encoded for each sequence and the different popularity distributions assumed for each set. For instance, for the sequence *GT_Fly* two types of scenes have been considered, one where the region of interest of the scene is at the right end of the multiview set (Fig. 3.8g) and another one where the major attention is expected to be at the center of the scene (Fig. 3.8h). Therefore, we have assumed the inverted exponential and the Gaussian distributions for the first and second sets of frames, respectively. A similar reasoning has been applied to the other sequences when selecting the different sets of frames and their associated popularity distribution. As the sequences *Book Arrival* and *Undo Dancer* are very homogeneous in time in terms of the position of the region of interest of the scene only one set of frames (frames 0-50) has been considered. Sample frames of the considered frame sets for each sequence are presented in Fig. 3.8. We also considered, for *Book Arrival* sequence and its unique set of frames, two different popularity distributions (Gaussian and uniform) to conclude about their impact on the PS and QPs selection.

Table 3.1 – Test conditions: encoded frame sets and popularity distribution for each test sequence.

Sequence	**Frame sets**	**View Pop. Distribution**
Poznan_Hall2	0-50	Exponential
	100-150	Gaussian
	150-200	U-Quadratic
Pantomime	0-50	Gaussian
	350-400	Inverted exponential
Book Arrival	0-50	Gaussian
	0-50	Uniform
GT_Fly	0-50	Inverted exponential
	125-175	Gaussian
Undo Dancer	0-50	Gaussian



(a) *Poznan_Hall2* sequence, coded view 0, frame 40.

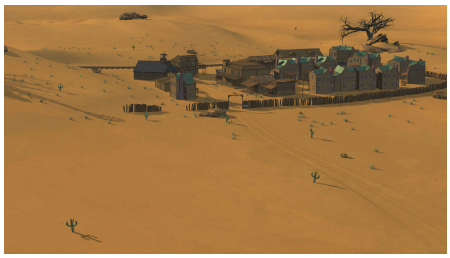
(b) *Poznan_Hall2* sequence, coded view 0, frame 200.

(c) *Pantomime* sequence, coded view 37, frame 1.

(d) *Pantomime* sequence, coded view 37, frame 370.

(e) *Book Arrival* sequence, coded view 8, frame 50.

(f) *Undo Dancer* sequence, coded view 1, frame 56.



(g) *GT_Fly* sequence, coded view 3, frame 1.

(h) *GT_Fly* sequence, coded view 3, frame 135.

Figure 3.8 – Content characteristics examples for the frame sets for each test sequence: (a) and (b) *Poznan_Hall2*, (c) and (d) *Pantomime*, (e) *Book Arrival*, (f) *Undo Dancer* and (g) *GT_Fly*.

3.6.2 Storage and Transmission Constraints

Given the different sequence characteristics, the best PS and associated texture and depth maps QPs have been found for various scenarios defined in terms of bandwidth and storage capacity. These scenarios are specified in Table 3.2 for each sequence under consideration. The defined TR_{max} and CR_{max} values were chosen in order to have a good video quality in terms of PSNR (30-40 dB). These values are different for the various sequences due to the particular content characteristics and image size. Regarding the allocation of the texture and depth coding rate, $CR_{t,max}$ and $CR_{d,max}$, we empirically found the appropriate ratio of the rate that provided the lowest expected distortion, as defined in Eq. (3.9). For instance, for the *Book Arrival* sequence the best percentage of rate allocated to the depth, CR_d, would be around 40% of the available bitrate budget. These values are consistent with the texture and depth maps rate allocation results available in the literature [71] [72], where they observe that the optimal bitrate ratio is significantly different depending on the sequence characteristics.

Table 3.2 – Test scenarios: bandwidth and storage capacity for each sequence.

Sequence	TR_{max} [Mbps]	$CR_{t,max}$ [Mbps]	$CR_{d,max}$ [Mbps]	CR_{max} [Mbps]
Poznan_Hall2	1	2	2	4
Pantomime	1.8	4.5	1	5.5
Book Arrival	0.7	1	0.7	1.7
GT_Fly	3.7	5.5	1.3	6.8
Undo Dancer	3	5	1	6

3.6.3 Results and Analysis

In Table 3.3 and 3.4, the optimal PSs and associated texture and depth maps QPs are shown for each sequence and set of frames when MVC and 3D-HEVC are used as codecs, respectively. We compare the performance of our proposed algorithm with the exhaustive search (ES) approach, which guarantees to find the global optimal PS, this means the PS minimizing the distortion while fulfilling the storage and bandwidth constraints. In the exhaustive search approach, at each stage of our graph, all the PSs and possible QPs are evaluated, while in our optimization algorithm only the PSs in each sub-stage are considered. Due to the content similarity and fixed view popularity distribution, the PS_g^* and Q_g^* found for all GOPs, of each frame set, were always the same. Therefore, in Table 3.3 and 3.4 only one PS_g^* and Q_g^* are shown per frame set and sequence.

The comparison between the proposed greedy algorithm and the ES approach is done here in terms of the Lagrangian cost as specified in (3.18), and the computational complexity, measured as CPU execution time. We use the normalized difference of the Lagrangian, $\Delta\mathscr{L}$, and the difference of execution time, ΔT, both in percentage. In particular, $\Delta\mathscr{L} = (\mathscr{L}_G - \mathscr{L}_{ES})*$

Table 3.3 – MVC greedy and exhaustive search solutions: results and performance comparison.

Sequence	**Frame sets**	**ES** (q_t, q_d)	**Greedy** (q_t, q_d)	$\Delta\mathscr{L}$, ΔT [%]
Poznan_Hall2	0-50	IIBIBPP	IIBIBPP	0, 71
		(36, 41)	(36, 41)	
	100-150	PBIIIBP	PBIIIBP	0, 71
		(37, 41)	(37, 41)	
	150-200	IIBPBIP	IIBPBIP	0, 72
		(37, 42)	(37, 42)	
Pantomime	0-50	PPPPIIPPPP	PPPPIIPPPP	0, 65
		(35, 34)	(35, 34)	
	350-400	PPPPPPPPII	PPPPPPPPII	0, 64
		(36, 34)	(36, 34)	
Book Arrival	0-50	PIIPP	PIIPP	0, 42
		(33, 35)	(33, 35)	
	0-50	PIPIP	PPIIP	2.2, 42
		(33, 36)	(33, 36)	
GT_Fly	0-50	PPPII	PPPII	0, 42
		(39,33)	(39,33)	
	125-175	PPPII	PPPII	0, 41
		(39,33)	(39,33)	
Undo Dancer	0-50	PPPII	PPPII	0, 42
		(35,27)	(35,27)	

$100/\mathscr{L}_G$ and $\Delta T = (T_{ES} - T_G) * 100/T_{ES}$, where the indexes ES and G are used to differentiate the Lagrangian and execution time obtained with exhaustive search and with our proposed greedy algorithm, respectively. The closer $\Delta\mathscr{L}$ is to zero, the closer the obtained PS solution is to the optimal solution in terms of RD performance. Moreover, the closer ΔT is to 100%, the larger is the complexity reduction obtained with the proposed algorithm compared to exhaustive search.

In general, the results obtained with the 3D-HEVC codec (Table 3.4) are very similar to the ones obtained with the MVC codec (Table 3.3), which shows how our proposed selection algorithm is independent of the specific codec used. The differences are due to the higher efficiency of the 3D-HEVC codec compared with MVC, obtaining PSs with lower optimal $Q = (q_t, q_d)$, and to the limitations of the 3D-HEVC reference software HTM. In the 3D-HEVC software version considered only 2 or 3 views can be simultaneusly coded, which limits the possible PSs as

Table 3.4 – 3D-HEVC greedy and exhaustive search solutions: results and performance comparison.

Sequence	**Frame sets**	**ES** (q_t, q_d)	**Greedy** (q_t, q_d)	$\Delta\mathscr{L}, \Delta T$ [%]
Poznan_Hall2	0-50	IIBPIBP	IIBPIBP	0, 50
		(33, 40)	(33, 40)	
	100-150	PBIIIBP	PBIIIBP	0, 50
		(34, 40)	(34, 40)	
	150-200	IIBPPBI	IIBPPBI	0, 51
		(34, 40)	(34, 40)	
GT_Fly	0-50	PPIII	PPIII	0, 57
		(28, 26)	(28, 26)	
	125-175	PPIPI	PPIPI	0, 44
		(29, 26)	(29, 26)	
Undo Dancer	0-50	PPIPI	PPIPI	0, 43
		(28, 26)	(28, 26)	

at least one key view should be available for every 3 coded views. For instance, in the case of $|\mathscr{V}| = 5$ the only two possible PSs with 3D-HEVC with one key view are: PBIBP and PPIPP. On the other hand, the MVC reference software provides more freedom when selecting the number and position of the key views.

As it can be seen from Table 3.3 and 3.4, the proposed algorithm is able to identify the global optimal PS ($\Delta\mathscr{L} = 0\%$) or near-optimal PS ($\Delta\mathscr{L} = 2.2\%$) with a complexity reduction of up to 72%, in comparison with the ES algorithm. The variation of the complexity reduction with the sequences is due to the number of coded views considered and the number of key views we are able to allocate, given the *CR* and *TR* constraints. The larger the number of coded views and allocated key views, the larger the complexity reduction is, as the number of PSs considered with our algorithm, compared with the ones considered with the ES approach, gets smaller. This is the case of *Poznan_Hall2* sequence, where our algorithm achieves a lower complexity reduction when 3D-HEVC is used compared to when MVC is used, as the possible PSs are fewer with the 3D-HEVC codec than with the MVC codec.

In general, we can observe an alignment of the optimal PSs with the popularity models, where for both the greedy and the ES algorithms, the chosen PSs allocate the key views to the most popular viewpoint positions. For instance, for the *Book Arrival* sequence, and the same set of frames, different allocations of the key views are proposed for the two popularity models considered, namely Gaussian and uniform. This is not so obvious for the *GT_Fly* and *Undo Dancer* sequence, where for all the view popularity distributions the optimal key views take the

lateral position in the multiview set. This is due to the non-uniform distribution of the coded and virtual views. For instance, when the MVC codec is used, the optimal chosen key views are the two coded views 5 and 9, which serve as reference views to render the virtual views considered. To render virtual views $\{6,7,8\} \in \mathscr{W}$ coded views 5 and 9 are needed as reference views, while virtual view $4 \in \mathscr{W}$ requires coded view 5 as the right reference view. Therefore, since six ($\{4,5,6,7,8,9\}$) out of nine available views for user request need coded views 5 and/or 9, it is expected that they should be independently encoded, as they contribute with most of the transmission bitrate.

Different from common PSs in the multiview compression literature, the best PSs, shown in Table 3.3 and 3.4, have more than one key view. This solution results from the trade-off between minimizing the transmission rate (associated to PSs with less interview dependencies) and maximizing the compression efficiency (associated to PSs with more interview dependencies). These results indicate that a pure compression efficiency objective is not ideal in IMVS systems. Note that in the case where there is an infinite bandwidth constraint, the optimal PSs will tend to maximize the inter-view dependencies proposing solutions with only one view independently encoded. Differently, if there is an infinite storage constraint, then the optimal PSs will tend to maximize the number of views independently encoded.

Though experiments have been done with the available data sets, which have a limited number of views or a small navigation range, similar results are expected in real IMVS applications where a large number of views should be available for user request and distant views considerably differ in their scene content. Note also that, predifined PSs may be used for further encoding scenarios by modeling different datasets according to their content. This would decrease the complexity associated to perform the proposed optimization algorithm for each video sequence, and therefore it could be used for live streaming cases.

3.7 Conclusions

In this chapter, we have proposed an algorithm that efficiently selects a near-optimal interview PS and associated texture and depth QPs, at the GOP level, when the MVD data format is used for IMVS systems. We consider an IMVS system where storage capacity and transmission rate are limited resources. While the search space is a priori quite big, our algorithm is able to reduce the set of relevant PSs and reduce the search complexity without significant RD performance penalty. To evaluate the performance of the proposed algorithm, the multiview video coding standards MVC and 3D-HEVC have been considered and simulation results have shown that the global optimal or near-optimal PS can be obtained with the proposed algorithm, while the associated complexity is considerably reduced (up to 72% of complexity reduction compared to an exhaustive search approach). Given a unique bandwidth constraint, we find the PS to encode all the views that will eventually be transmitted to the users, but the solutions does not adapt to heterogeneous networks. In the next chapter, we propose an adaptive solution where users of an IMVS system have different access link bandwidth

capabilities.

4

Optimal Layered Representation for Adaptive IMVS

4.1 Introduction

In IMV, the quality of the rendered views in the navigation window depends on the quality of the captured views and on their relative distance, as the distortion of a virtual view tend to increase with the distance to the views used as references in the view synthesis process. This means that, in the ideal case, all the captured views encoded at the highest possible rate, would be transmitted to all the clients. However, in practice, resource constraints prevent the transmission of all the views. In particular, clients may have different access link bandwidth capabilities, and some of them may not be able to receive all the captured views. In this context, it becomes important to find adaptive solutions for interactive multiview video streaming (IMVS) systems that adapt to the capabilities of the clients.

In this chapter, we consider the problem of jointly determining which views to transmit and at what encoding rate, such that the expected rendering quality in the navigation window is maximized under relevant resource constraints. In particular, we consider the scenario illustrated in Fig. 4.1, where a set of views are captured from an array of time-synchronized cameras. For each captured view, both a texture and a depth map are available, so that intermediate virtual viewpoints can eventually be synthesized. The set of captured and virtual views defines the navigation window available for client viewpoint request. Clients are clustered in groups according to their bandwidth capabilities; for instance, in Fig. 4.1 only one client per cluster is illustrated for three groups with 1Mbps, 5Mbps and 10Mbps bandwidth

constraints. Then, the set of captured views are organized in layers or subsets of views to be transmitted to the different groups of clients in order to maximize the overall navigation quality. With a layered organization of the captured views in the navigation window, we aim at offering a progressive increase of the rendering quality. Indeed, the quality of the navigation improves with the number of layers (subset of views) that clients are able to receive. In the example of Fig. 4.1, three layers or subsets of views are formed as: $\mathcal{V}_1 = \{1, 6\}$, $\mathcal{V}_2 = \{4\}$ and $\mathcal{V}_3 = \{2, 3, 5\}$. Depending on the clients' bandwidth capabilities, they receive the views in $\mathcal{V}_1$, or in $\mathcal{V}_1$ and $\mathcal{V}_2$, or in $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_3$. In particular, the client with the lowest bandwidth capability (*i.e.*, the client with a mobile phone) is able to receive only the subset of views $\mathcal{V}_1$ in the first layer, and needs to synthesize the rest of the views. On the other hand, the client with the highest bandwidth capability (*i.e.*, the client with a TV), is able to receive all the views, and therefore reaches the highest navigation quality.

We formulate an optimization problem to jointly determine the optimal arrangement of views in layers along with the coding rate of the views, such that the expected rendering quality is maximized in the navigation window, while the rate of each layer is constrained by network and clients capabilities. We show that this combinatorial optimization problem is NP-hard, meaning that it is computationally difficult and there are not known algorithm that optimally solves the problem in polynomial time. We then propose a globally optimal solution based on the dynamic-programing (DP) algorithm. As the computational complexity of this algorithm grows with the number of layers, a greedy and lower complexity algorithm is proposed, where the optimal subset of views and their coding rates are computed successively for each layer by a DP-based approach. The results show that our greedy algorithm achieves a close-to-optimal performance in terms of total expected distortion, and outperforms a distance-based view and rate selection strategy used as a baseline algorithm for layer construction.

This chapter is organized as follows. First, the main characteristics of the layered IMV representation are outlined in Section 4.2 where also our optimization problem is formulated. Section 4.3 describes the optimal and greedy views selection and rates allocation algorithms for our layered multiview representation. Section 4.4 presents the experimental results that show the benefits of the proposed solution and the conclusions are outlined in Section 4.5.

4.2 Framework and Problem Formulation

We consider the problem of building a layered multiview video representation in an IMVS system, where the clients are heterogeneous in terms of bandwidth capabilities. In this section, we first describe the most relevant characteristics of the IMVS system. Then, we formulate our optimization problem.

4.2.1 Network and IMVS Model

In this work, we denote as $\mathcal{V}^o = \{1, 2, \ldots, V\}$ the ordered set of captured views from an array of synchronized cameras defining a navigation window for the clients. Each camera compresses the recorded view before transmitting it over the network. We assume that there is no communication between the cameras, so each camera encodes its images independently of the other cameras, which is common in numerous novel applications ranging from surveillance to remote sensing. For each captured view in $\mathcal{V}^o$, both a texture and a depth map are available so that users can eventually synthesize new viewpoints using DIBR techniques. At the decoder side each client can reconstruct a view at any position in the discrete set $\mathcal{U} = \{1, 1+\delta, \cdots, V\}$; with δ as the minimum distance between consecutive views in the navigation window.

We consider a population of heterogeneous clients requesting camera views from the IMVS system, such that they can freely navigate within the navigation window defined by the views in $\mathcal{V}^o$. Due to resource constraints in practical systems, it is not possible to transmit all the camera views in $\mathcal{V}^o$ to all the clients. Therefore, we propose a *layered multiview representation*, where clients are clustered according to their bandwidth capabilities and the set of views transmitted to each group of clients are carefully selected, so that their navigation quality is maximized. This means that, given a set of received views, an intermediate view u can be left uncoded at the encoder, if $u \in \mathcal{V}^o$, or simply be a virtual view, if $u \in \mathcal{U} \setminus \mathcal{V}^o$. In both cases, view u can be synthesized at the decoder using the two surrounding available encoded views at the user side, v_L and v_R, where $v_L < u < v_R$ for $v_L, v_R \in \mathcal{V}^o$.

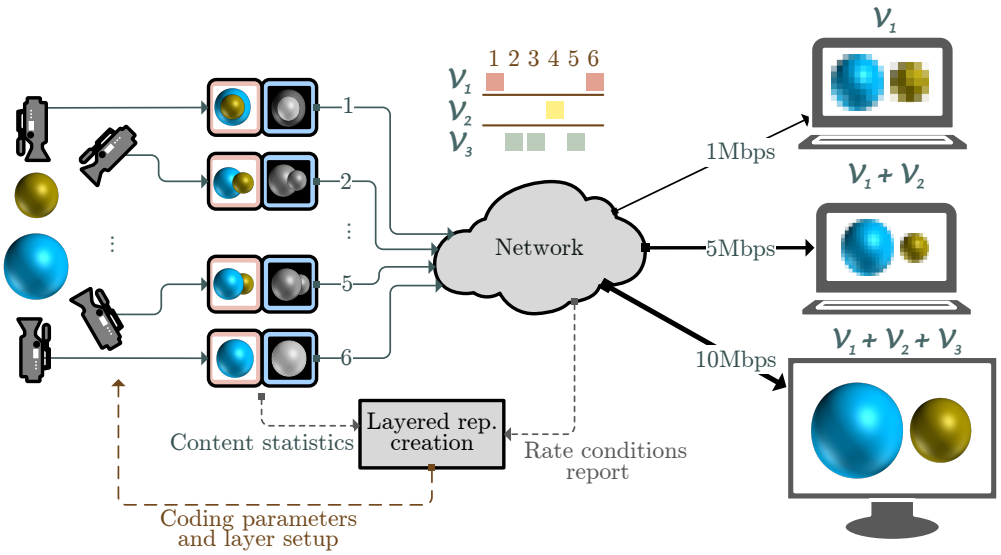


Figure 4.1 – Illustration of an IMVS system with 6 camera views and 3 heterogeneous clients. The optimization is done by the *layered representation creation* module considering three layers defined by the set of views $\{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$.

4.2.2 Layered Multiview Video Representation Model

We give now some details on the proposed *layered multiview representation*. The views in $\mathcal{V}^o$, are organized into layered subsets $\mathcal{V} = \{\mathcal{V}_1, \cdots, \mathcal{V}_C\}$ to offer a progressive increase of the visual navigation quality with an increasing number of layers. In particular, the finite set of cameras $\mathcal{V}^o$ is divided in C layers such that $\mathcal{V}_1 \cup \mathcal{V}_2 \cup \cdots \cup \mathcal{V}_C \subseteq \mathcal{V}^o$, with $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, i \neq j$. The number of layers C corresponds to the number of subsets of heterogeneous clients grouped according to their bandwidth capabilities. As a requirement, a client cannot decode a view in $\mathcal{V}_c$ without receiving the views in $\mathcal{V}_{c-1}$, meaning that $\mathcal{V}_1$ and $\mathcal{V}_C$ are the most and the least important subsets, respectively. This means that clients with very low bandwidth capabilities may only receive the views in the first layer ($\mathcal{V}_1$), and need to synthesize the missing viewpoints. On the other hand, clients with higher bandwidth capabilities receive more layers, which leads to a lower rendering distortion as the distance between reference views decreases, hence the view synthesis is of better quality. In addition, we denote by $\mathcal{V}_1^c = \bigcup_{l=1}^{c} \mathcal{V}_l = [v_1, \cdots, v_N]$ the ordered subset of N views in the first c layers, for $c \leq C$, and by $\mathcal{R}_1^c = \{r_{v_1}, \cdots, r_{v_N}\}$ as the set of rates chosen to encode the selected views in $\mathcal{V}_1^c$, where $r_{v_i} \in \mathcal{R}^o$ and $\mathcal{R}^o$ is the set of all possible rates for a given encoder. When $c = C$, we simply denote $\mathcal{V}_1^c = \mathcal{V}$ and $\mathcal{R}_1^c = \mathcal{R}$. As mentioned before, we assume that view synthesis with DIBR is done by using a right and left reference views. Therefore, the leftmost and rightmost views of the navigation window need to be transmitted in the first layer, $v_1 = 1$ and $v_N = V$.

Formally, the quality of the interactive navigation when the views from the c most important layers are received and decoded can be defined as:

$$D_c(\mathcal{V}_1^c, \mathcal{R}_1^c) = \sum_{\substack{u \in \mathcal{U}, \\ v_L, v_R \in \mathcal{V}_1^c, \\ r_{v_L}, r_{v_R} \in \mathcal{R}_1^c}} p(u)\, d_u(v_L, v_R, r_{v_L}, r_{v_R}) \tag{4.1}$$

$$\text{with,} \qquad v_L = \min_{\substack{v \in \mathcal{V}_1^c, \\ v < u}} |v - u| \qquad v_R = \min_{\substack{v \in \mathcal{V}_1^c, \\ v \geq u}} |v - u|$$

where v_L and v_R are the closest right and left reference views to view u among the views in $\mathcal{V}_1^c$, and d_u is the distortion of view u, when it is synthesized using v_L and v_R as reference views, encoded at rates r_{v_L} and r_{v_R}, respectively. Finally, $p(u)$ is the view popularity factor describing the probability that a client selects view $u \in \mathcal{U}$ for navigation [66]. We assume that $p(u)$, depends on the popularity of the views, due to the scene content, but it is independent of the view previously requested by the client. Note that, $D_c \geq D_{c+1}$, since each camera views subset or layer provides a refinement of the navigation quality experienced by the client.

4.2.3 Problem Formulation

We now formulate the optimization problem for the allocation of coded views in layers and their rate allocation in order to maximize the expected navigation quality for all IMVS clients. More specifically, the problem is to find the optimal subset of captured views from the set of available views $\mathcal{V}^o$ that should be allocated to each of the C layers $\mathcal{V}^* = \{\mathcal{V}_1^*, \cdots, \mathcal{V}_C^*\}$ and optimal coding rate of each selected view in $\mathcal{V}^*$, $\mathscr{R}^* = \{r_{v_1}^*, \cdots, r_{v_N}^*\}$, such that the expected distortion of the navigation is minimized for all the clients, while the bandwidth constraint per layer, $\mathbf{B} = [B_1, \cdots, B_C]$, is satisfied. This bandwidth constraint is associated to the bandwidth capabilities of each clients cluster. The optimization of the number of layers and of the rate constraints of the layers due to clients' bandwidth capabilities is out of the scope of this paper. Formally, the optimization problem can be written as:

$$\min_{\mathcal{V}_1^c, \mathscr{R}_1^c} \sum_{c=1}^{C} q(c)\, D_c(\mathcal{V}_1^c, \mathscr{R}_1^c) \qquad \text{such that,} \qquad \sum_{v_i \in \mathcal{V}_1^c} r_{v_i} \leq B_c, \qquad \forall c \in \{1, \cdots, C\} \tag{4.2}$$

where $q(c)$ stands for the proportion of clients that are able to receive the first c layers $\mathcal{V}_1^C$, namely clients with rate capability larger than B_c but lower than B_{c+1}. The distortion D_c is given in Eq. (4.1). We finally assume that the depth maps are all encoded at the same high quality, as accurate depth information is important for view synthesis. In practice, the coding rate of depth maps is much smaller than the rate of the texture information, even when compressed at high quality [83]. In the above problem formulation, the rate of encoded views can be formally written as $r_{v_i} = r_{v_i}^t + r_{v_i}^d$, with $r_{v_i}^t$ and $r_{v_i}^d$ as the rate of the texture and depth information of view v_i, respectively. For the sake of clarity, and without loss of generality, we assume in the following that $r_{v_i} = r_{v_i}^t$, due to the low rate contribution of the compressed depth maps compared with the texture information.

Note that in the case where the navigation domain is too large, we can split it in sub-domains in order to ensure a particular navigation quality, notably for users receiving only the first layers. In this case, we would have a set $\mathcal{V}^o$ defining the limits of each sub-domain and then the problem posed in (4.2) is independently solved for each of them.

4.2.4 NP-Hardness Proof

We now prove that the optimization problem in (4.2) is NP-hard, by reducing it to a well-known NP-complete problem, the *Knapsack* problem. The Knapsack problem is a combinatorial problem that can be characterized as follows:

Settings – Non-negative weights $w_1, w_2, \cdots, w_V$, profits $c_1, c_2, \cdots, c_V$, and capacity W.
Problem – Given a set of items, each with a weight and a profit, find a subset of these items such that the corresponding profit is as large as possible and the total weight is less than or

equal to W.

We now consider a simplified instance of our problem in (4.2) and consider only one layer and a unique rate value for each captured view. Intuitively, if the problem is NP-hard for this simplified case it will also be NP-hard for the full optimization problem. We reduce this simplified problem from the Knapsack problem. First, we map each weight w_v to a view rate r_{v_i}. Then, when a view v_i is considered as a reference view for the corresponding layer, the profit is quantified by the distortion reduction that it brings in total, denoted here as $\theta(v_i)$, where $\theta(v_i) = D_c(\mathcal{V}_1^c, \mathscr{R}_1^c) - D_c(\overline{\mathcal{V}}_1^c, \overline{\mathscr{R}}_1^c)$, for $\overline{\mathcal{V}}_1^c = [\mathcal{V}_1^c\ v_i]$ and $\overline{\mathscr{R}}_1^c = [\mathscr{R}_1^c\ r_{v_i}]$. However, the profit $\theta(v_i)$ of each view is not independent from the content of current and previous layers, as it is the case for each object in the Knapsack problem. The profit depends on the views that have been already selected as reference views in the layer, meaning $\mathcal{V}_1^c$. This increases the complexity of the view selection and rate allocation problem compared to the classic Knapsack problem. Therefore, if the problem is NP-hard when profits $\theta(v_i)$ are independent of the layer content, then it will be NP-hard for our simplified problem. Then, assuming an independent profit for each view, our simplified problem can be rewritten as:

Settings – Rates of the possible reference views $r_1, r_2, \cdots, r_V$, independent profit for each view $\theta(1), \theta(2), \cdots, \theta(V)$, and bandwidth capacity B_c.
Problem – Given a set of views, each with a rate and a profit, find the subset of views such that the distortion reduction is as large as possible and the total rate is less than or equal to B_c.

This reduced problem is equivalent to the Knapsack problem. Hence, this proves that our original optimization problem is at least as hard as the Knapsack problem. Therefore, our problem in (4.2) is NP-hard.

4.3 Proposed Optimization Algorithms

To tackle the problem in (4.2), we propose first an algorithm that solves the optimization optimally. Second, we present a reduced complexity algorithm that finds a locally optimal solution working on a layer by layer basis, with an average quality performance close to the optimal algorithm.

4.3.1 Optimal Algorithm

To obtain an optimal solution to the problem in (4.2), we propose a dynamic programming (DP) algorithm that solves problems by breaking them down in subproblems and combining their solutions. The subproblems are solved only once, and their solutions are stored in a DP table to be used in the multiple instances of the same subproblem [54]. To develop a DP algorithm from the problem defined in (4.2), we first need to identify the structure of the problem and how it can be decomposed. We start with the following observations:

1. *Decomposition in the view domain* – We first observe that the distortion D_c in Eq. (4.1) can be computed by parts and recursively. In particular, we can write:

$$\begin{aligned} D_c(\mathcal{V}_1^c, \mathscr{R}_1^c) &= \Delta_c(v_1, r_{v_1}) + \sum_{n=1}^{N-1} \Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) \\ &= \Delta_c(v_1, r_{v_1}) + \Delta_c(v_1, v_2, r_{v_1}, r_{v_2}) + D_c(\mathcal{V}_1^c \setminus v_1, \mathscr{R}_1^c \setminus r_{v_1}) \end{aligned} \tag{4.3}$$

where, $\Delta_c(v_1, r_{v_1})$ denotes the distortion of view $v_1 \in \mathcal{V}_1^c$ encoded at rate r_{v_1}, and it can be written as:

$$\Delta_c(v_1, r_{v_1}) = q(v_1)\, d_{v_1}(v_1, r_{v_1}) \tag{4.4}$$

The distortion between consecutive views v_n and v_{n+1} in $\mathcal{V}_1^c$, compressed at rates r_{v_n} and $r_{v_{n+1}}$, respectively, should account for the distortion of the synthesized views, $v_n < u < v_{n+1}$, with $u \in \mathcal{V}^o \setminus \mathcal{V}_1^c$ and coded view v_{n+1}. This distortion is denoted as $\Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}})$, which is defined as:

$$\Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) = \sum_{\substack{v_n < u \le v_{n+1}, \\ v_n, v_{n+1} \in \mathcal{V}_1^c, \\ r_{v_n}, r_{v_{n+1}} \in \mathscr{R}_1^c}} p(u)\, d_u(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) \tag{4.5}$$

2. *Decomposition in the layer domain* – Given a multiview layered representation of C layers, we denote as $\phi_c^C(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}})$ the expected distortion between reference views v_n and v_{n+1} encoded at rates r_{v_n} and $r_{v_{n+1}}$, when v_n and v_{n+1} are the closest reference views in $\mathcal{V}_1^c$ and $\mathcal{V}_1^C = \mathcal{V}$ (i.e, no intermediate views are added between views v_n and v_{n+1} from layer c to layer C). The distortion $\phi_c^C(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}})$ can be expressed as:

$$\begin{aligned} \phi_c^C\left(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}\right) &= \sum_{l=c}^{C} q(l)\, \Delta_l(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) \\ &= q(c)\, \Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) + \phi_{c+1}^C\left(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}\right) \end{aligned} \tag{4.6}$$

As users receiving higher layers need also to receive all the previous layers for optimal quality improvement, the reference views in layer c become available for any layer $l > c$. This means that, the distortion difference for clients in layers c and $c+1$ simply depends on the improvement provided by views in $\mathcal{V}_{c+1}$. In other words, the expected distortion can be computed iteratively.

Let $\Phi_c^C\left(v_n, v, r_{v_n}, r_v, \overline{\mathbf{B}}_c^C\right)$ be the minimum expected distortion between reference views v_n and v encoded at rates r_{v_n} and r_v, when the remaining rate budget for each layer, including the subset of camera views that can be added between v_n and v, is $\overline{\mathbf{B}}_c^C = \left[\overline{B}_c, \overline{B}_{c+1}, \cdots, \overline{B}_C\right]$. Based on the above observations, (4.3) and (4.6), this minimum distortion can be recursively defined as follows:

$$\Phi_c^C\left(v_n, v, r_{v_n}, r_v, \overline{\mathbf{B}}_c^C\right) = \min_{\substack{v_n < v_{n+1} \leq v \mid v_{n+1} \in \mathcal{V}^o \setminus \mathcal{V}_1^{c-1} \\ 0 \leq r_{v_{n+1}} \leq \overline{B}_c \mid r_{v_{n+1}} \in \mathscr{R}^o \\ 0 \leq \mathbf{b}_{c+1}^C \leq \overline{\mathbf{B}}_{c+1}^C}} q(c)\Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) +$$
$$\Phi_{c+1}^C\left(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}, \mathbf{b}_{c+1}^C\right) + \Phi_c^C\left(v_{n+1}, v, r_{v_{n+1}}, r_v, \overline{\mathbf{B}}_c^C - \begin{bmatrix} r_{v_i} \\ \mathbf{b}_{c+1}^C \end{bmatrix}\right) \tag{4.7}$$

In each recursive call, (4.7) finds the optimal $\{v_{n+1}, r_{v_{n+1}}\}$ and eventually $\mathbf{b}_{c+1}^C$, that minimizes the distortion between views v_n and v given the bit budget $\overline{\mathbf{B}}_c^C$ between layer c and C. The first term in (4.7) corresponds to the layer distortion Δ_c between views v_n and v_{n+1}, as defined in (4.5). The second term defines the minimum distortion between views v_n and v_{n+1} from layer $c+1$ to layer C, when the rate constraint assigned to each layer is $\mathbf{b}_{c+1}^C = [b_{c+1}, \cdots, b_C]$, for $\mathbf{b}_{c+1}^C \leq \overline{\mathbf{B}}_{c+1}^C$. Finally, the third term is associated to the minimum expected distortion for clients receiving the views from layer c to C, between views v_{n+1} and v when the rate constraint is $\overline{\mathbf{B}}_c^C - \begin{bmatrix} r_{v_{n+1}} \\ \mathbf{b}_{c+1}^C \end{bmatrix}$. Given the first view is always selected in the first layer, (4.7) can be solved via the following initial call for the first layer:

$$\min_{\substack{0 \leq r_1 \leq B_1 \\ r_1 \in \mathscr{R}^o}} q(1)\,\Delta_1(1, r_1) + \Phi_2^C(1, r_1, \emptyset) + \Phi_1^C\left(1, V, r_1, r_V, \mathbf{B}_1^C - \begin{bmatrix} r_1 \\ \emptyset \end{bmatrix}\right) \tag{4.8}$$

If $v_{n+1} = v$, meaning it is the rightmost view between the two reference views, then the last recursive term in (4.7), $\Phi_c^C(.)$, is not needed. Similarly, if $c = C$, meaning that the current layer is the last considered layer, then the first recursive term in (4.7), $\Phi_{c+1}^C(.)$, is not needed. The three terms in (4.7) are illustrated in Fig. 4.2a for views from layer c to layer $c+1$.

A DP-table is used to store the solution of each sub-problem $\Phi_c^C\left(v_n, v, r_{v_n}, r_v, \overline{\mathbf{B}}_c^C\right)$ for a given layer c. Each solution is stored in the entry $[v_n][v][r_{v_n}][r_v][\overline{\mathbf{B}}_c^C]$ of the c DP-table. Hence, the complexity of the algorithm is bounded by the size of each DP-table $\mathcal{O}(V^2R^2B^C)$, where R is the size of the set $\mathscr{R}^o$ of available rates and B as the larger budget value in $\mathbf{B}$. The complexity is also determined by the complexity of calculating each table entry: $\mathcal{O}(VRB^C)$. Thus, given C DP-tables, the total complexity of the algorithm is $\mathcal{O}\left(CV^3R^3B^{2C}\right)$, which is exponential with

the number of layers C.

4.3.2 Greedy Algorithm

The computational time to solve the optimization problem with the above DP algorithm is exponential and rapidly grows with the number of available layers. Therefore, we propose a greedy approximate solution where the optimization problem defined in (4.2) is solved successively for each layer, starting from the first layer. When solving the optimization problem for each layer, the optimal reference views are selected from the full set of captured views when optimizing the first layer, while for the following layers, the solution is restricted to the views that have not been selected as reference views in the previous layers. However, the intuition behind this greedy algorithm is that, in our system, the lowest layers are necessary to most of the clients, for which our greedy algorithm tends to be close to optimal. Therefore, it is expected that our greedy algorithm leads to an effective solution in terms of overall expected distortion. Formally, the greedy algorithm considers the following optimization problem for each layer c:

$$\min_{\mathcal{V}_c^*, \mathscr{R}_c^*} q(c)\, D_c(\mathcal{V}_1^c, \mathscr{R}_1^c) \qquad \text{such that,} \qquad \sum_{v_i \in \mathcal{V}_c} r_{v_i} \leq B_c \tag{4.9}$$

where, $\mathscr{R}_c$ stands for the set of coding rates of the views selected as reference views in layer c.

To obtain an approximate solution, meaning the optimal solution in each particular layer given the set of available reference views, we propose a DP algorithm inspired on the algorithm presented in Section 4.3.1. Let $\Phi_c\left(v_n, r_{v_n}, \overline{B}_c\right)$ be the minimum expected distortion at layer c between reference views v_n, encoded at rate r_{v_n}, and the last view of the set $\mathcal{V}^o$, V, as it is always selected. The remaining rate budget $\overline{B}_c$ is available for selecting new views in layer c between the given reference views, v_n and V. This optimal solution is again a recursive function that finds the optimal $\{v_{n+1}, r_{v_{n+1}}\}$, with $v_n < v_{n+1} < V$, minimizing $\Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}})$ and the optimal solution Φ_c in the remaining set of views between v_{n+1} and V. This can be formally written as:

$$\Phi_c\left(v_n, r_{v_n}, \overline{B}_c\right) = \min_{\substack{v_n < v_{n+1} \leq V \,|\, v_{n+1} \in \mathcal{V}^o \setminus \mathcal{V}_1^{c-1} \\ 0 \leq r_{v_{n+1}} \leq \overline{B}_c \,|\, r_{v_{n+1}} \in \mathscr{R}^o}} \Delta_c(v_n, v_{n+1}, r_{v_n}, r_{v_{n+1}}) + \Phi_c\left(v_{n+1}, r_{v_{n+1}}, \overline{B}_c - r_{v_{n+1}}\right) \tag{4.10}$$

A DP algorithm implements the recursive formulation in (4.10) to determine the optimal allocation of views in layer c, given the allocation in previous layers. In each recursive call, the

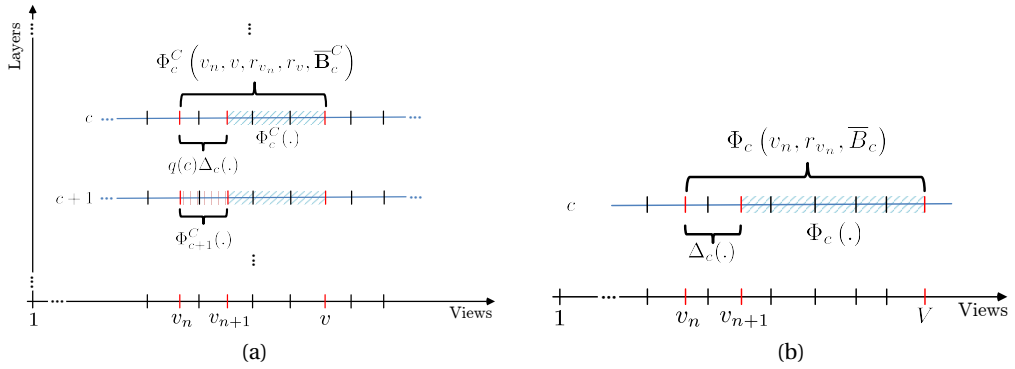


Figure 4.2 – (a) Illustration of the optimal algorithm with the three terms from (4.7) for views in layers c and $c+1$ that are in the recursive evaluation of $\Phi_c^C\left(v_n, v, r_{v_n}, r_v, \overline{\mathbf{B}}_c^C\right)$. (b) Greedy algorithm illustration with the two terms from (4.10) for views in layer c and the recursive function $\Phi_c\left(v_n, r_{v_n}, \overline{B}_c\right)$.

optimal v_{n+1} and corresponding $r_{v_{n+1}}$ that minimizes the distortion between v_n and V given the available rate budget $\overline{B}_c$ is found. The algorithm runs for each layer successively, starting from the first layer. Similarly to (4.8), given that the first view in $\mathcal{V}^o$ is always selected, (4.10) can be solved via the following initial call in each layer c:

$$\min_{\substack{0\le r_{v_1}\le \overline{B}_c \\ r_1\in\mathscr{R}^o}} \Delta_c(1, r_1) + \Phi_c\left(1, r_1, \overline{B}_c - r_1\right) \tag{4.11}$$

In (4.10), if $v_{n+1} = V$, then the recursive term is not needed. In Fig. 4.2b, the two terms in (4.10) are illustrated for views in a general layer c.

Finally, following a similar analysis than the one followed in Section 4.3.1, the algorithm in (4.10) has a complexity $\mathcal{O}\left(CV^2R^2B\right)$. The size of the DP-table in this case is VRB, and the complexity due to filling each entry of the table is $\mathcal{O}(VR)$. Moreover, the algorithm should run C times, one time for each layer. By solving every layer successively in the greedy algorithm, we are able to remove the exponential dependency with the number of layers, in the complexity of the algorithm; hence to seriously reduce the overall computational complexity of the optimal optimization algorithm. Note that, as for the complexity estimated for the optimal algorithm in Section 4.3.1, this estimated complexity is not related to the encoding process followed in each camera or the decoding and view synthesis process done at the decoder side. It corresponds to the complexity of solving the problem defined in (4.2) with the corresponding proposed algorithm.

4.4 Performance Assessment

This section presents the test conditions and performance results obtained in different scenarios when the search of the optimal subset of coded views per layer and rate allocation per view is performed with the algorithms proposed in this paper. We study the optimal allocation in different settings and compare it to the solution of a baseline camera distance-based solution.

4.4.1 General Test Conditions

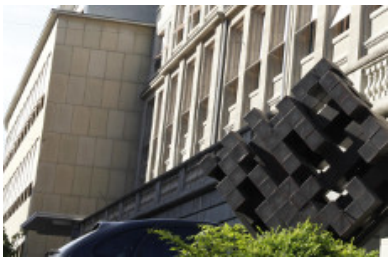
We consider four different data sets for evaluating the performance of our optimization algorithms. We first study the performance on two multiview video datasets, *Ballet* (1024 × 768, 15Hz) [2] and *Undo Dancer* (1920 × 1080, 25Hz) [79]. Though the main target of this work is on video delivery, we also consider two multiview image datasets, *Statue* (2622 × 1718) and *Bikes* (2676 × 1752) [4], due to the relatively high quality of their depth maps compared with the ones available in multiview video sequences. Multiview image experiments permits to appreciate the benefits of our solution in allocating resources based on scene content properties. The 3D-HEVC reference software HTM 6.2 [75] has been used to encode jointly texture and depth maps in each dataset. The views are encoded independently and temporal prediction is used for each view in the video sequences. The depth maps are encoded at high quality (we set a quantizer scale factor of QP=25 for the depth maps), while a set of different rate values $\mathscr{R}^o$ is considered for encoding the texture information. For each sequence, the following conditions have been considered:

- *Statue* – A total of $|\mathscr{V}^o| = 7$ captured views and $|\mathscr{U}| = 10$ equally spaced rendered views are considered. In this dataset, the cameras are horizontally arranged with a fixed distance between neighboring cameras of 5.33mm. We have chosen the ten available views to have a separation of at least 26.65mm between pair of views, such that $\mathscr{U} = \{50\ 55\ 60\ 65\ 70\ 75\ 80\ 85\ 90\ 95\}$ and $\mathscr{V}^o = \{50\ 55\ 65\ 70\ 80\ 85\ 95\}$, in terms of view indexes in the dataset.

- *Bikes* – A total of $|\mathscr{V}^o| = 7$ and $|\mathscr{U}| = 7$ captured and rendered views are considered, respectively. In this dataset, the cameras are horizontally arranged with a spacing of 5mm. As for *Statue* dataset, to increase the distance between available views, we have chosen the available views by fixing the minimum distance between views to be 25mm. In detail, the seven views correspond to the views $\mathscr{V}^o = \mathscr{U} = \{10\ 20\ 25\ 30\ 35\ 40\ 50\}$, in terms of dataset indexes.

- *Ballet* – A total of $|\mathscr{V}^o| = 7$ captured views and $|\mathscr{U}| = 8$ rendered views are considered. The views follow a circular arrangement and correspond to $\mathscr{V}^o = \{0\ 1\ 2\ 4\ 5\ 6\ 7\}$ and $\mathscr{U} = \{0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\}$, regarding the view indexes in the dataset.

- *Undo Dancer* – A total of $|\mathscr{V}^o| = 5$ captured views and $|\mathscr{U}| = 9$ equally spaced rendered views are considered. The cameras for this sequence are horizontally arranged with a



(a) *Statue* multiview image, view 1.

(b) *Bikes* multiview image, view 1.

(c) *Ballet* multiview sequence, view 1, frame 1.

(d) *Undo Dancer* multiview sequence, view 3, frame 1.

Figure 4.3 – Content characteristics example for a frame of each considered multiview image and video dataset: (a) *Statue*, (b) *Bikes*, (c) *Ballet*, (d) *Undo Dancer*.

fixed distance of 20 cm between neighboring views. They correspond to the captured views $\mathcal{V}^o = \{1\,2\,3\,5\,9\}$ and the nine available views for rendering $\mathcal{U} = \{1\,2\,3\,4\,5\,6\,7\,8\,9\}$, in terms of dataset indexes.

In Fig. 4.3 a frame of each of the considered multiview video and image dataset is illustrated.

The distortion of any synthesized view u at the decoder depends on the quality of the reference views used for synthesis, namely v_L and v_R, and on their distance to the synthesized view. For the simulations, we use a distortion model, which considers these two factors in estimating the distortion of the synthetic view d_u as:

$$d_u(v_L, v_R) = (1-\alpha)\left(d_{v_1^t}(v_L, v_R) + d_{v_1^d}(v_L, v_R)\right) + (1-\gamma)\alpha\left(d_{v_2^t}(v_L, v_R) + d_{\hat{v}_2^d}(v_L, v_R)\right) + \gamma\alpha\mathscr{I} \tag{4.12}$$

where, $d_{v_i^t}$ and $d_{v_i^d}$, for $i \in \{1,2\}$, denote the average distortion per pixel for the texture and the depth map of the first and second views that are used as references for view synthesis, where $v_i \in \{v_L, v_R\}$. The parameters α and γ are respectively the proportion of disoccluded pixels in the projection of the first reference view and in the projections of both reference views in the

DIBR view synthesis. Their values depend only on the scene geometry and they are obtained from the depth maps of the reference views. Finally, the average distortion per pixel in the inpainted areas is denoted by $\mathscr{I}$, which is assumed to take a constant value that only depends on the scene content. This distortion model is further explained in Appendix A.

Throughout this section, performance results are shown in terms of the expected distortion that we denote here as $\overline{D}$, and it is defined as $\sum_{c=1}^{C} q(c)D_c$, with D_c in (4.1). Although simulations are done using the distortion model in (4.12), the distortion shown as $\overline{D}$ in this section, is estimated after using the 3D-HEVC encoder to encode the selected reference views and after synthesizing the missing views using DIBR.

In the rest of this section, we carry out simulations for different system settings to evaluate the performance of our greedy and our optimal algorithms presented in Sections 4.3.1 and 4.3.2. We compare their performance to those of a baseline algorithm, which selects a subset of coded views per layer such that the average distance between reference and synthetic views is minimized in each layer.

4.4.2 Greedy vs. Optimal Algorithm

In this section, we compare the performance of both the optimal and greedy algorithms proposed in Sections 4.3.1 and 4.3.2. Due to the exponential complexity of our optimal algorithm, a small discrete set of available rates $\mathscr{R}^o$ to encode the texture information is used and only two layers are considered in the layered multiview representation, which means that the clients are clustered in only two groups depending on their bandwidth capabilities.

We consider two different distributions for the proportion of clients that subscribe to each layer. In particular, we set q = [0.5 0.5], when the first half of the clients can only get $\mathscr{V}_1$ and the second half get both $\mathscr{V}_1$ and $\mathscr{V}_2$, and we set q = [0.1 0.9], when most of the clients have high bandwidth capabilities and only 10% of them can only get the views in the first layer, $\mathscr{V}_1$. We also assume that all the views in $\mathscr{U}$ have the same probability of being requested, which results in a uniform view probability distribution p.

The results are presented in Table 4.1, where the set of views per layer $\mathscr{V}^*$ and the expected distortion $\overline{D}$ are shown for each considered data set. The rate constraint per layer B_c and the set of available rates $\mathscr{R}^o$ to encode the texture information for each of the considered datasets are given in Table 4.1. The views selected by each algorithm in each layer are given in terms of the rate, $\mathscr{V}_c = \{r_1, \cdots, r_v, \cdots r_V\}$, where $r_v = 0$ means that the view is not transmitted in that particular layer and $r_v > 0$ means that the view is encoded at rate r_v in the corresponding layer. The indexes of the views correspond to the views arrangement in the set of captured views $\mathscr{V}$.

It can be seen from the results in Table 4.1 that the same optimal set of views per layer $\mathscr{V}^*$ has been chosen for both the greedy and optimal solutions when a uniform distribution of q for the clients is assumed. The same results have been obtained for values of $q(1)$ higher than 0.5, but they are not presented here due to space restrictions. When $q(2)$ increases, meaning

Table 4.1 – Comparison of the optimal and greedy algorithms in terms of view selection and rate allocation $\mathcal{V}^*$ and average distortion $\overline{D}$.

Sequence & Settings	q	**Optimal** $\mathcal{V}^*$	$\overline{D}(dB)$	**Greedy** $\mathcal{V}^*$	$\overline{D}(dB)$
Statue	[0.5 0.5]	$\mathcal{V}_1 = \{2\,0\,2\,0\,2\,0\,2\}$	38.22	$\mathcal{V}_1 = \{2\,0\,2\,0\,2\,0\,2\}$	38.22
B_c = 8Mb		$\mathcal{V}_2 = \{0\,2\,0\,2\,0\,4\,0\}$		$\mathcal{V}_2 = \{0\,2\,0\,2\,0\,4\,0\}$	
$\mathscr{R}^o = \{0\,2\,4\}$Mb	[0.1 0.9]	$\mathcal{V}_1 = \{2\,0\,2\,0\,2\,0\,2\}$	39.45	$\mathcal{V}_1 = \{2\,0\,2\,0\,2\,0\,2\}$	39.45
		$\mathcal{V}_2 = \{0\,2\,0\,2\,0\,4\,0\}$		$\mathcal{V}_2 = \{0\,2\,0\,2\,0\,4\,0\}$	
Bikes	[0.5 0.5]	$\mathcal{V}_1 = \{1.5\,1.5\,0\,2\,0\,1.5\,1.5\}$	37.13	$\mathcal{V}_1 = \{1.5\,1.5\,0\,2\,0\,1.5\,1.5\}$	37.13
B_c = 8Mb		$\mathcal{V}_2 = \{0\,0\,2\,0\,2\,0\,0\}$		$\mathcal{V}_2 = \{0\,0\,2\,0\,2\,0\,0\}$	
$\mathscr{R}^o = \{0\,1.5\,2\}$Mb	[0.1 0.9]	$\mathcal{V}_1 = \{2\,0\,2\,0\,2\,0\,2\}$	38.48	$\mathcal{V}_1 = \{1.5\,1.5\,0\,2\,0\,1.5\,1.5\}$	38.11
		$\mathcal{V}_2 = \{0\,2\,0\,2\,0\,2\,0\}$		$\mathcal{V}_2 = \{0\,0\,2\,0\,2\,0\,0\}$	
Ballet	[0.5 0.5]	$\mathcal{V}_1 = \{0.3\,0\,0\,0.3\,0\,0\,0.3\}$	38.56	$\mathcal{V}_1 = \{0.3\,0\,0\,0.3\,0\,0\,0.3\}$	38.56
B_c = 1 Mbps		$\mathcal{V}_2 = \{0\,0.3\,0.3\,0\,0.3\,0\,0\}$		$\mathcal{V}_2 = \{0\,0.3\,0.3\,0\,0.3\,0\,0\}$	
$\mathscr{R}^o = \{0\,0.25\,0.3\}$Mbps	[0.1 0.9]	$\mathcal{V}_1 = \{0.3\,0\,0\,0.3\,0\,0\,0.3\}$	40.37	$\mathcal{V}_1 = \{0.3\,0\,0\,0.3\,0\,0\,0.3\}$	40.37
		$\mathcal{V}_2 = \{0\,0.3\,0.3\,0\,0.3\,0\,0\}$		$\mathcal{V}_2 = \{0\,0.3\,0.3\,0\,0.3\,0\,0\}$	
Undo Dancer	[0.5 0.5]	$\mathcal{V}_1 = \{0.5\,0\,0\,1\,0.5\}$	36.87	$\mathcal{V}_1 = \{0.5\,0\,0\,1\,0.5\}$	36.87
B_c = 2 Mbps		$\mathcal{V}_2 = \{0\,1\,1\,0\,0\}$		$\mathcal{V}_2 = \{0\,1\,1\,0\,0\}$	
$\mathscr{R}^o = \{0\,0.5\,1\}$Mbps	[0.1 0.9]	$\mathcal{V}_1 = \{0.5\,0\,0\,1\,0.5\}$	36.98	$\mathcal{V}_1 = \{0.5\,0\,0\,1\,0.5\}$	36.98
		$\mathcal{V}_2 = \{0\,1\,1\,0\,0\}$		$\mathcal{V}_2 = \{0\,1\,1\,0\,0\}$	

that the second layer is transmitted to a larger group of clients, the greedy algorithm shows its sub-optimality. For instance, when $q(2) = 0.9$ the optimal solution is not obtained by the greedy algorithm for the *Bikes* dataset; instead, the same $\mathcal{V}^*$ solution as for $q(1) = q(2) = 0.5$ is computed. This sub-optimality is due to the fact that, in our greedy algorithm, the problem is solved successively for each layer, starting from the first layer. This means that the optimal solution $\mathcal{V}^*$ does not depend on the probability distribution q of clients requesting each layer. Therefore, the solution $\mathcal{V}^*$ for each dataset is the same for any distribution q; it only affects the expected distortion $\overline{D}$. This successive approach of our greedy algorithm also means that the first layer is prioritized, where the layer $c = 1$ always has an optimal set of views independently of the other layers. This explains the good performance of the greedy algorithm when the first layer has high probability of being transmitted alone, *i.e.*, high value of $q(1)$. Nevertheless, even when the second layer is transmitted to a larger group of clients, $q(2) = 0.9$, the greedy algorithm shows a good performance, presenting an optimal solution for three of the four datasets considered in our experiments. This good performance of the greedy algorithm can be explained by the fact that the first layer is always received by all the clients, independently of the probability distribution q of clients requesting each layer. Therefore, optimizing the allocation of the views in the first layer is never really bad, which further justifies the design of our greedy algorithm. In addition, it has a lower complexity compared with the optimal algorithm, as demonstrated in Section 4.3.2. Therefore, for the rest of the paper we only consider the greedy algorithm and we compare it with a baseline solution for view selection

and rate allocation.

4.4.3 Greedy Algorithm Performance

After showing the good performance of our greedy algorithm in the previous section, we now study its performance in different scenarios and compare it with a baseline algorithm, namely *distance-based view selection solution* [84]. In this algorithm, the views in each layer are selected such that the distance between encoded and synthesized views is minimized. Views are encoded at the same rate in each layer and the rate per view and the number of views are chosen such that the available bandwidth per layer is used to its maximum. Layers are filled in successive order, as for our greedy algorithm.

The algorithms are compared in different settings where the layer rate constraint and view popularity effects are evaluated. A total of four layers are considered in all the simulations presented in this section, representing four groups of clients that are clustered depending on their bandwidth capabilities. Note that, since we do not consider our optimal algorithm in these simulations, we are able to increase the set of available coding rates $\mathscr{R}^o$ for each dataset and the number of layers in the multiview layered representation, compared with the experiments in Section 4.4.2.

Layer Rate Constraint Variations

In this subsection the greedy algorithm is compared with the distance-based solution in terms of the expected distortion when varying the layer rate constraint. We use an illustrative layer rate distribution that follows a linear relationship: $B_c = x \times c + y$. By varying the values of x and y, we can study the performance of the view selection algorithm in different settings. The corresponding results are presented in Table 4.2, where the solution from the greedy algorithm outperforms the distance-based solution in terms of the expected distortion $\overline{D}$ in 4 out of 6 experiments. On the other two cases, the same result is obtained by both algorithm. The performance gain obtained with our greedy algorithm is mainly due to its rate allocation capability compared to the homogeneous rate assignment in the distance-based algorithm. The non-uniform rate allocation characteristic of our greedy algorithm permits the fully use of the available rate per layer, allocating more bits to views used as references in the view synthesis process; *e.g.*, for layer $c = 2$ with *Undo Dancer* sequence when $\{x\ y\} = \{0.5\ 0.5\}$. In Fig. 4.4, we show the visual quality and the Y-PSNR value of the view 20 from the *Bikes* image dataset given the results shown in Table 4.2 when $\{x\ y\} = \{0.5\ 4\}$ for our greedy and the distance-based algorithms. In particular, in Fig. 4.4a we show the synthesized imaged using the reference views 10 and 30, encoded at rates $r_{10} = r_{30} = 1.5Mb$ of *Bikes* image dataset, which corresponds to the visual quality achieved by users receiving only layer 1. Then, in Fig. 4.4b and 4.4c, we show the image encoded at rates $r_{20} = 2Mb$ and $r_{20} = 1.5Mb$, as selected by our gredy algorithm and the distance-based solution, respectively. These are the images consumed at layer 2 and 3 when view 20 is requested by the users. As it can be seen the visual

Table 4.2 – Comparison of the greedy and distance-based algorithm for different layer rate constraints.

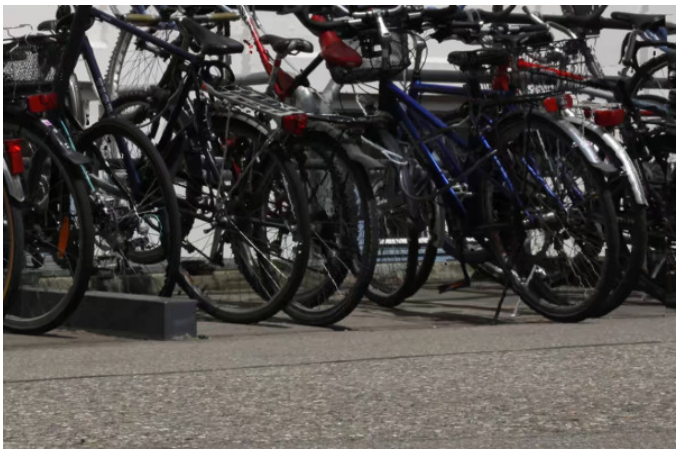
Sequence & Settings	**Rate** **{x, y}**	**Greedy** $\mathcal{V}^*$	$\overline{D}(dB)$	**Distance-based** $\mathcal{V}^*$	$\overline{D}(dB)$
Bikes $\mathscr{R}^o$ = {0 1 1.5 2 2.5 2.7} Mb	{2, 2}	$\mathcal{V}_1$ = {2 0 0 0 0 0 2} $\mathcal{V}_2$ = {0 1.5 1.5 0 1.5 1.5 0} $\mathcal{V}_3$ = {0 0 0 2.7 0 0 0}	33.75	$\mathcal{V}_1$ = {2 0 0 0 0 0 2} $\mathcal{V}_2$ = {0 1.5 1.5 0 1.5 1.5 0} $\mathcal{V}_3$ = {0 0 0 2.7 0 0 0}	33.75
	{0.5, 4}	$\mathcal{V}_1$ = {1.5 0 0 1.5 0 0 1.5} $\mathcal{V}_2$ = {0 2 0 0 1.5 1.5 0} $\mathcal{V}_3$ = {0 0 2.7 0 0 0 0}	35.33	$\mathcal{V}_1$ = {1.5 0 0 1.5 0 0 1.5} $\mathcal{V}_2$ = {0 1.5 0 0 1.5 1.5 0} $\mathcal{V}_3$ = {0 0 2.7 0 0 0 0}	35.22
Ballet $\mathscr{R}^o$ = {0 0.15 0.18 0.20 0.25 0.3} Mbps	{0.25, 0.25}	$\mathcal{V}_1$ = {0.15 0 0 0.2 0 0 0.15} $\mathcal{V}_2$ = {0 0.2 0.25 0 0 0.3 0} $\mathcal{V}_3$ = {0 0 0 0 0.3 0 0}	39.35	$\mathcal{V}_1$ = {0.25 0 0 0 0 0 0.25} $\mathcal{V}_2$ = {0 0 0.25 0.25 0 0.25 0} $\mathcal{V}_3$ = {0 0.3 0 0 0.3 0 0}	37.73
	{0.2, 0.1}	$\mathcal{V}_1$ = {0.15 0 0 0 0 0 0.15} $\mathcal{V}_2$ = {0 0.25 0 0.25 0 0 0} $\mathcal{V}_3$ = {0 0 0.3 0 0 0.3 0} $\mathcal{V}_4$ = {0 0 0 0 0.3 0 0}	37.95	$\mathcal{V}_1$ = {0.15 0 0 0 0 0 0.15} $\mathcal{V}_2$ = {0 0 0.25 0 0.25 0 0} $\mathcal{V}_3$ = {0 0.3 0 0 0 0.3 0} $\mathcal{V}_4$ = {0 0 0 0.3 0 0 0}	37.90
Undo Dancer $\mathscr{R}^o$ = {0 0.25 0.5 0.75 1 1.25} Mbps	{0.5, 0.5}	$\mathcal{V}_1$ = {0.5 0 0 0 0 0.5} $\mathcal{V}_2$ = {0 0 0 0.5 1 0} $\mathcal{V}_3$ = {0 1.25 0 0 0 0}	36.48	$\mathcal{V}_1$ = {0.5 0 0 0 0 0.5} $\mathcal{V}_2$ = {0 0 0.75 0.75 0 0} $\mathcal{V}_3$ = {0 1.25 0 0 0 0}	36.35
	{0.25, 0.75}	$\mathcal{V}_1$ = {0.5 0 0 0 0 0.5} $\mathcal{V}_2$ = {0 0 0 1.25 0 0} $\mathcal{V}_3$ = {0 0.75 0.75 0 0 0}	36.63	$\mathcal{V}_1$ = {0.5 0 0 0 0 0.5} $\mathcal{V}_2$ = {0 0 0 1.25 0 0} $\mathcal{V}_3$ = {0 0.75 0.75 0 0 0}	36.63

quality increased with the number of layers received.

In general, the distance-based view selection solution shows to be relatively close to the optimal solution, where most of the selected views in each layer are almost equally spaced. This can be also seen by comparing the visual quality of Fig. 4.4b and Fig. 4.4c. This is due to the small change in content among different views, which is due to the small distance between the cameras and/or the low scene complexity in most of the available datasets. Nevertheless, these experiments have shown that a simple distance-based solution with a uniform rate allocation among the selected views in each layer, is not ideal as it cannot take into account the actual content of the scene, contrarily to our algorithm.

View Popularity Distribution Variations

Now we compare our greedy algorithm with the distance-based solution when views have different popularities. The results are shown for an exponential popularity distribution, where the leftmost and rightmost views in the set of captured views $\mathcal{V}$ are the most and the least popular view, respectively. Note that a different popularity distribution could have been used. The results are presented in Table 4.3, where the optimal set of views per layer $\mathcal{V}^*$ and the total



(a) Greedy and distance-based algorithm - view synthesized by the users in layer 1 (PSNR=29.3 dB).

(b) Distance-based algorithm - view decoded by the users in layers 2 and 3 (PSNR=35.6 dB).

(c) Greedy algorithm - view decoded by the users in layers 2 and 3 (PSNR=36.8 dB).

Figure 4.4 – View 20 of *Bikes* dataset as rendered for users in layers 1, 2 and 3 using the greedy and the distance-based algorithm.

Table 4.3 – Greedy and distance-based solutions comparison for an exponential view popularity distribution.

Sequence	**Greedy**		**Distance-based**	
& Settings	$\mathcal{V}^*$	$\overline{D}(dB)$	$\mathcal{V}^*$	$\overline{D}(dB)$
Statue	$\mathcal{V}_1 = \{4\,0\,2\,0\,0\,0\,0\,2\}$		$\mathcal{V}_1 = \{4\,0\,0\,0\,0\,0\,0\,4\}$	
$B_c = 8$ Mb	$\mathcal{V}_2 = \{0\,4\,0\,0\,0\,4\,0\,0\}$	37.13	$\mathcal{V}_2 = \{0\,0\,4\,0\,0\,4\,0\,0\}$	36.87
$\mathcal{R}^o = \{0\,2\,4$	$\mathcal{V}_3 = \{0\,0\,0\,4\,4\,0\,0\,0\}$		$\mathcal{V}_3 = \{0\,0\,0\,4\,4\,0\,0\,0\}$	
$5\,6\,8\}$ Mb	$\mathcal{V}_4 = \{0\,0\,0\,0\,0\,0\,4\,0\}$		$\mathcal{V}_4 = \{0\,4\,0\,0\,0\,0\,4\,0\}$	
Bikes	$\mathcal{V}_1 = \{2\,0\,0\,0\,0\,0\,1.5\}$		$\mathcal{V}_1 = \{1.5\,0\,0\,0\,0\,0\,1.5\}$	
$B_c = 3.5$Mb	$\mathcal{V}_2 = \{0\,1.5\,2\,0\,0\,0\,0\}$	35.49	$\mathcal{V}_2 = \{0\,0\,1.5\,0\,1.5\,0\,0\}$	33.89
$\mathcal{R}^o = \{0\,1\,1.5$	$\mathcal{V}_3 = \{0\,0\,0\,2\,1.5\,0\,0\}$		$\mathcal{V}_3 = \{0\,1.5\,0\,0\,0\,1.5\,0\}$	
$2\,2.5\,2.7\}$ Mb	$\mathcal{V}_4 = \{0\,0\,0\,0\,0\,2.7\,0\}$		$\mathcal{V}_4 = \{0\,0\,0\,2.7\,0\,0\,0\}$	
Ballet	$\mathcal{V}_1 = \{0.25\,0\,0\,0\,0\,0\,0.25\}$		$\mathcal{V}_1 = \{0.25\,0\,0\,0\,0\,0\,0.25\}$	
$B_c = 0.5$Mbps	$\mathcal{V}_2 = \{0\,0\,0.3\,0.2\,0\,0\,0\}$	39.29	$\mathcal{V}_2 = \{0\,0\,0.25\,0\,0.25\,0\,0\}$	39.13
$\mathcal{R}^o = \{0\,0.15\,0.18$	$\mathcal{V}_3 = \{0\,0.3\,0\,0\,0.2\,0\,0\}$		$\mathcal{V}_3 = \{0\,0.25\,0\,0\,0\,0.25\,0\}$	
$0.20\,0.25\,0.3\}$ Mbps	$\mathcal{V}_4 = \{0\,0\,0\,0\,0\,0.3\,0\}$		$\mathcal{V}_4 = \{0\,0\,0\,0.3\,0\,0\,0\}$	
Undo Dancer	$\mathcal{V}_1 = \{0.75\,0\,0\,0\,0.5\}$		$\mathcal{V}_1 = \{0.5\,0\,0\,0\,0.5\}$	
$B_c = 1.25$Mbps	$\mathcal{V}_2 = \{0\,0.75\,0\,0.5\,0\}$	36.57	$\mathcal{V}_2 = \{0\,0\,0\,1.25\,0\}$	36.48
$\mathcal{R}^o = \{0\,0.25\,0.5$	$\mathcal{V}_3 = \{0\,0\,1.25\,0\,0\}$		$\mathcal{V}_3 = \{0\,0\,1.25\,0\,0\}$	
$0.75\,1\,1.25\}$ Mbps	$\mathcal{V}_4 = \{0\,0\,0\,0\,0\}$		$\mathcal{V}_4 = \{0\,1.25\,0\,0\,0\}$	

expected distortion $\overline{D}$ are shown for the greedy and distance-based solutions. The settings for the different sequences are specified in the Table 4.3. The total expected distortion $\overline{D}$ is calculated assuming a uniform distribution of the proportion clients accessing each layer, meaning $q = [0.25\ 0.25\ 0.25\ 0.25]$, for the four layers. The results show that the solution from the greedy algorithm outperforms the distance-based solution in terms of the total expected distortion. This is due to the fact that the distance-base solution does not consider neither the popularity distribution of the views nor an optimized rate allocation among the views. In particular, in the greedy algorithm the views close to the leftmost view (the most popular views) are selected in the first layers, to ensure that most of the clients receive the most popular views and therefore enjoy a higher expected navigation quality. Similar conclusions can be drawn when considering other view popularities distributions.

An alternative presentation of the gain of our greedy algorithm is shown in Fig. 4.5. A bar plot illustrates the expected quality (Y-PSNR) of our greedy algorithm (GA) and of the distance-based approach (DBA) for the four considered layers in these simulations. We consider the *Bikes* and *Ballet* datasets, with the same settings as the ones of the results in Table 4.3. In addition, we have included horizontal lines representing the average quality of each algorithm

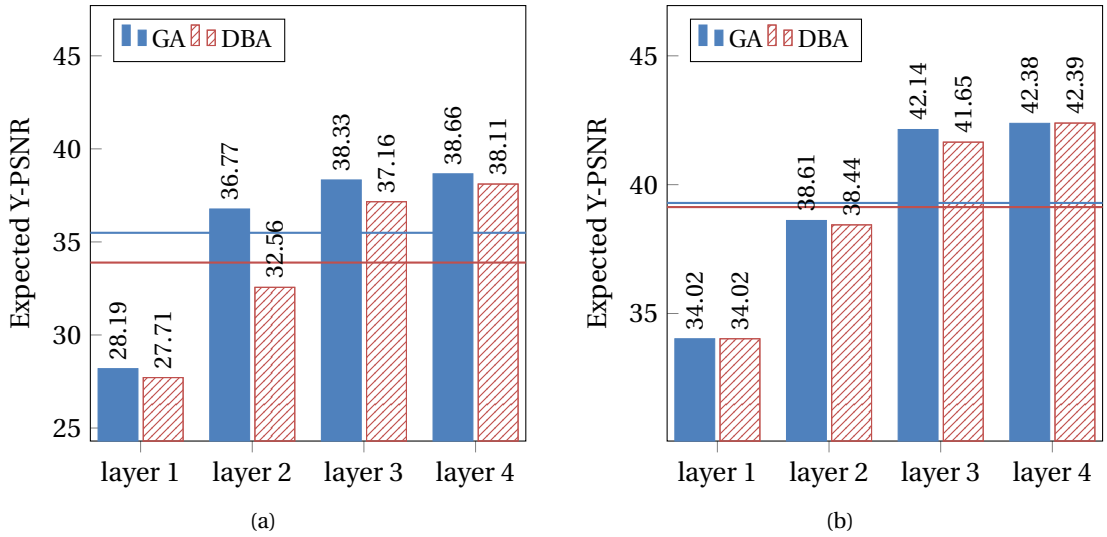


Figure 4.5 – Layer-by-layer Y-PSNR(dB) for the conditions specified in Table 4.3, for (a) *Bikes* and (b) *Ballet* datasets when comparing our greedy algorithm (GA) and the distance-based algorithm (DBA) performance.

across the whole client population (the four client clusters), using the same bar color. The distortion is calculated with the views received in the current layer and in all the previous layers, as clients subscribed to a particular layer receive all the views up to that layer. Therefore, for both approaches, the overall quality increases as the layer index increases since clients are able to receive more views. Note that, in general, our greedy algorithm outperforms the distance-based approach, achieving the highest average quality. In the case of the *Ballet* sequence, we can see however that the group of clients receiving up to layer $c = 4$ enjoy a slightly higher quality with the distance-based approach than with the greedy algorithm. This is due to the fact that in the fourth layer all the reference views are selected and most of them are encoded at the highest possible rate for the distance-based approach, as it was the only option for the algorithm to fully use the available bandwidth and have a uniform rate allocation among the selected views. However, this view and rate selection of the distance-based solution only favors clients in the last cluster (highest bandwidth capabilities). In fact, the overall performance for the *Ballet* sequence is better for our greedy algorithm, as for the first layers the view selection and rate allocation offer a higher quality to the first three group of clients.

4.5 Conclusions

We have proposed a novel adaptive transmission solution that jointly selects the optimal subsets of views and the rate allocation per view for an adaptive transmission in IMVS applications. We consider a system where the network is characterized by clients with heterogeneous bandwidth capabilities, and we aim to minimize their expected navigation distortion. To do so,

clients are clustered according to their bandwidth capabilities and the different camera views are distributed in layers to be transmitted to the different groups of users in a progressive way, such that the clients with higher capabilities receive more layers (more views), hence benefiting of a better navigation quality. We have formulated an optimization problem to jointly determine the optimal arrangement of views in layers along with the coding rate of the views, such that the expected rendering quality is maximized in the navigation window, while the rate of each layer is constrained by network and clients capabilities. To solve this problem, we have proposed an optimal algorithm and a greedy algorithm with a reduced complexity, both based on dynamic-programming. It has been shown through simulations that the proposed algorithms are able to reduce the navigation distortion in a IMVS system. In addition, our greedy algorithm has close-to-optimal performance and outperforms a baseline algorithm based on an equidistant view distribution with an uniform rate allocation among the selected views in each layer. Our results show that, considering the client capabilities and their preferences in navigation, as well as the 3D scene content, is key in the design of an effective adaptive transmission solution for IMVS systems. However, due to complexity reasons, we do not allow here inter-view prediction, which would increase the compression efficiency of the presented solution. Therefore, in the next chapter, we focus on the rate allocation problem. Based on the solutions presented in this chapter, we propose a general reduced-complexity rate allocation algorithm that allows inter-view prediction in multiview video settings.

5

Optimal Lagrange Multiplier Values for Constrained Rate Allocation Problems

5.1 Introduction

In rate allocation problems, the goal is to optimally distribute a rate budget among a set of coding units [1]. As there is a common trade-off between lossy compression rate and resulting distortion, the optimal rate distribution is normally the one minimizing the distortion given a rate budget.

To control the rate allocation, the selection of the quantization parameters (QPs) and the sub-sampling of coding units are commonly used strategies. By modifying the QP of each coding unit we are able to control the coding rate and the quality, where a larger QP results in a low bit-rate (low quality) and a smaller QP results in a high bit-rate (high quality). In some cases, not encoding and transmitting some units can even be a better strategy in terms of rate-distortion (R-D) trade-off value than using a large QP. In this case, missing units can still be rendered at the decoder side in a post-processing step using available surrounding coded units; *e.g.*, using depth-image based rendering (DIBR) [86], [17] method to synthesize missing views in multiview scenarios when texture and depth maps are available. By skipping coding units, the QP assigned to the remaining units can be decreased as the available rate budget gets distributed on a smaller number of coding units, increasing the quality of the

[1] The term *coding unit* should not be confused with the term used by HEVC standard to denote a particular type of block after the frame partitioning [85]. Here it refers to images captured at a given time instant or from a given viewpoint, namely coded views in a multiview video or coded frames in a traditional monoview video.

coded units. However, the quality of the uncoded units that are reconstructed at the decoder depends on the distance to the coded units used as references and on their quality. Thus, the QP and the set of units skipped from encoding need to be jointly adjusted to reach optimal R-D performance.

In order to solve rate allocation problems, different approaches based on dynamic programing [87] or Lagrangian optimization [50], [51], [52] [48], [49] have been proposed. Methods based on dynamic programming provide optimal solutions; however the complexity is rather high as it basically compares all the possible solutions. Thus, rate allocation problems are usually solved by considering an unconstrained problem based on Lagrangian optimization, where a Lagrange multiplier λ is used to define a Lagrangian cost function in the form $D + \lambda R$, which permits to trade-off rate R and distortion D. There exists an optimal λ value that defines the best performance under constraints on either the rate or the distortion. Usually, the search for this optimal λ is done by swapping its value from an initial lower bound to an upper bound from a predetermined set of λ values, where a bisection search can be used to reduce the number of iterations. However, finding the optimal λ value is not guaranteed as the accuracy of the solution depends on the granularity of the search space. An algorithm to find the optimal Lagrange multiplier is proposed in [50] for the specific case of independently coded units with a fixed unit rate.

In this chapter, we study the rate allocation problem of finding the optimal subset of coding units and QPs in a multiview video scenario such that the expected distortion (*i.e*, view popularity-weighted distortion) among all the available units at the users is minimized. In particular, we propose an algorithm that finds the optimal Lagrange multiplier value with a minimum number of iterations. We consider the rate allocation problem introduced in Chapter 4, where for each layer the optimal subset of independently encoded views and QPs are optimized in multiview video given a layer rate constraint. We first review the complexity of the dynamic programming solution in Chapter 4, in the context of predictively coded units, and then we propose a reduced-complexity algorithm that is able to solve the rate allocation problem in polynomial time. We consider a general rate allocation problem with a Lagrangian formulation, where the Lagrange multipliers are optimally selected. Compared to [50], our solution is more general as the set of coding units are unknown (*i.e*, they need to be optimized) and they are predictively coded.

The rest of the chapter is organized as follows. In Section 5.2, we provide a brief description of the system model and we formulate our optimization problem for constrained rate allocation with predictively or dependent coded units. Then, in Section 5.3, we first show the high complexity of a dynamic programming algorithm for computing the optimal rate allocation for predictively encoded units. Then, by combining dynamic programming and a Lagrangian-based algorithm with an optimal search of the Lagrange multipliers we are able to reduce the complexity of the rate allocation solution. In Section 5.4, we propose an algorithm that finds the optimal Lagrange multiplier in a Lagrangian-based rate allocation problem. Simulation results, in Section 5.5, show the performance of the proposed rate allocation algorithm with

optimal Lagrange multiplier selection when units are independently or predictively coded for multiview and monoview video scenarios. We show that, our simple solution compares favorably to rate control solutions adopted in the reference softwares of current monoview and multiview video standards, namely HEVC [9] and 3D-HEVC [10]; with more complex rate control algorithms.

5.2 Rate Allocation Problem

In a classical rate allocation problem setup, the objective is to minimize the expected distortion of a set of coding units, which may be independently or differentially coded, subject to a single rate budget constraint. In this section, we first describe the system under consideration; then, we formulate a general constrained rate allocation problem for dependent coded units as a discrete optimization problem.

5.2.1 Multiview Video System

We consider a general coding scenario where we seek to allocate a total rate budget B to a set of V coding units. Specifically, let $\mathcal{V}^o = \{1, 2, \ldots, V\}$ be an ordered set of V coding units (*e.g.*, consecutive viewpoint images or views in a multiview sequence). We define $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$, where $\mathcal{V}^o \subseteq \mathcal{V}$, as the subset of N units, $N \leq V$, that are considered for coding. We assume that an intermediate unit v can be skipped at the encoder, and is estimated or synthesized at the decoder using the two surrounding coded units v_L and v_R, where $v_L < v < v_R$ and $v_L, v_R \in \mathcal{V}$. This implies that the boundary units cannot be synthesized at the decoder and are always selected for coding, *i.e.*, $v_1 = 1$ and $v_N = V$ are always coded.

Each unit $v_n \in \mathcal{V}$, selected for coding, is coded using a quantization parameter (QP) $q_{v_n} \in \mathcal{Q}$, where $\mathcal{Q}$ is the set of possible QPs for a given codec. Assuming that predictive coding is employed between units, unit v_n coded with a QP of q_{v_n} using as a predictor unit v_{n-1} coded with a QP of $q_{v_{n-1}}$ will result in the rate $r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$ and expected distortion $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$, where the user interaction behavior is characterized by a view popularity distribution used to penalize the distortion of each unit. Note that if v_{n-1} and v_n are not consecutive units in $\mathcal{V}^o$, then the intermediate units in $\mathcal{V} \setminus \mathcal{V}^o$ between v_{n-1} and v_n must be synthesized at the decoder. The expected distortion $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$ at the decoder must account for the distortion of all synthesized units in the range (v_{n-1}, v_n) as well as the coded distortion at v_n. Hence, the distortion $\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n})$ can be formally written as:

$$\Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) = \sum_{\substack{v_{n-1} < v \leq v_n \mid v \in \mathcal{V}^o \\ q_{v_{n-1}}, q_{v_n} \in \mathcal{Q}}} p(v)\, d_v(v_{n-1}, v_n, q_{v_{n-1}}, q_{v_n}) \tag{5.1}$$

where, $p(v)$ stands for the popularity or the probability of requesting/receiving unit v and

$d_v(.)$ stands for the distortion of unit v, $v \in \mathcal{V}^o$, synthesized using the reference units v_{n-1} and v_n coded using QPs $q_{v_{n-1}}$ and q_{v_n}, respectively. If $v = v_n$, then $d_v(.)$ corresponds to the distortion of coding unit v_n using unit v_{n-1} for prediction. Recall that the first coded unit in $\mathcal{V}$ is independently encoded; its distortion depends only on its own QP q_{v_1}. For this particular case, Eq. (5.1) can be re-written as:

$$\Delta_{v_1}(q_{v_1}) = p(v_1)\, d_{v_1}(q_{v_1}) \tag{5.2}$$

A more general definition of the distortion with predictive coding is also possible [51], where the rate and distortion depend on the QPs of all the previous coded units. However, due to complexity reasons, we assume here that the rate r_{v_n} and the distortion Δ_{v_n} depend only on the QP $q_{v_{n-1}}$ of the unit v_{n-1} used for prediction. This is a good approximation of the rate and distortion in practical predictive coding, as it has been shown in [49].

5.2.2 Problem Formulation

Given the above system model, our objective is to find the optimal subset of units $\mathcal{V} = \{v_1, v_2, \cdots, v_N\} \subseteq \mathcal{V}^o$ with their corresponding QPs $\mathbf{q} = [q_{v_1}, q_{v_2}, \cdots, q_{v_N}]$ such that the expected distortion at the decoder is minimized, subject to a global rate budget constraint B. The optimization problem can be defined as follows:

$$\begin{aligned} \min_{\mathcal{V},\mathbf{q}} \;& \Delta_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathcal{V}|} \Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) \\ \text{s.t.} \;& r_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathcal{V}|} r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) \leq B \end{aligned} \tag{5.3}$$

where $\Delta_{v_1}(q_{v_1})$ and $r_{v_1}(q_{v_1})$ are the distortion and rate for the first selected unit v_1, and $\Delta_{v_n}(.)$ and $r_{v_n}(.)$ are the distortion and rate for a predictively coded unit v_n, as described above.

Note that a special case of the problem defined in (5.3) is when units are independently coded. In this case, the units do not depend on previous coded units as temporal and/or inter-view prediction are ignored, i.e. $r_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) = r_{v_n}(q_{v_n})$. The problem posed in (5.3) is reduced to:

$$\min_{\mathcal{V},\mathbf{q}} \; \Delta_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathcal{V}|} \Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) \quad \text{s.t.} \quad \sum_{n=1}^{|\mathcal{V}|} r_{v_n}(q_{v_n}) \leq B \tag{5.4}$$

In this work, we assume that the problem is solved for each set of units (*i.e.*, a GOP in monoview video sequences or a set of views in a multiview video sequence), where all but the first unit are predictively encoded using an $IPP\cdots$ coding model. Meaning that the coding model is not optimized.

5.2.3 NP-Hardness Proof

We now prove that the optimization problem presented in (5.3) is NP-hard, by reducing it to a well-known NP-complete problem, the *Knapsack* problem. The Knapsack problem is a combinatorial problem that can be characterized as follows:

Settings – Non-negative weights $w_1, w_2, \cdots, w_V$, profits $c_1, c_2, \cdots, c_V$, and capacity W.
Problem – Given a set of items, each with a weight and a profit, find a subset of these items such that the corresponding profit is as large as possible and the total weight is less than or equal to W.

We now consider a simplified instance of our problem posed in (5.3), where each coding unit is associated to a unique QP value. Intuitively, if the problem is NP-hard for this simplified case it will also be NP-hard for the full optimization problem. We reduce this simplified problem from the Knapsack problem. First, we map each weight w_v to a unit QP q_{v_i}. Then, when an unit v_i is considered as a coded unit, the profit is quantified by the distortion reduction that it brings, denoted here as $\theta(v_i)$, where $\theta(v_i) = \Delta_{v_n}(v_{n-1}, q_{v_{n-1}}, q_{v_n}) - [\Delta_{v_i}(v_{n-1}, q_{v_{n-1}}, q_{v_i}) + \Delta_{v_n}(v_i, q_{v_i}, q_{v_n})]$, for v_n, v_{n-1} as consecutive coded units in $\mathcal{V}$ before considering coding unit v_i. However, differently from the Knapsack problem, the profit $\theta(v_i)$ is not independent of previously selected units. This increases the complexity of the unit selection and rate allocation problem compared to the classic Knapsack problem. Therefore, if the problem is NP-hard when the profit of a particular unit $\theta(v_i)$ is independent of the previously selected units, then it will be NP-hard for our simplified problem. Then, assuming an independent profit for each view, our simplified problem can be rewritten as:

Settings – QPs of the possible coding units $q_1, q_2, \cdots, q_V$, independent profit for each unit $\theta(1), \theta(2), \cdots, \theta(V)$, and bandwidth capacity B.
Problem – Given a set of units, each with an associated QP and profit, find the subset of coding units such that the distortion reduction is as large as possible and the total rate is less than or equal to B.

This reduced problem is equivalent to the Knapsack problem. Hence, this proves that our original optimization problem is at least as hard as the Knapsack problem. Therefore, our problem in (5.3) is NP-hard.

5.3 Lagrangian Optimization

We first present an algorithm based on dynamic programming (DP) that returns the optimal solution to the problem in (5.3). We then show that, the complexity of this algorithm is exponential, and thus we propose a polynomial-time alternative by relaxing the problem constraint with a Lagrangian problem formulation.

5.3.1 Constrained DP Algorithm

To obtain an optimal solution to the problem (5.3), we propose a DP algorithm that recursively divides the original problem into sub-problems. Whenever a sub-problem is solved, its solution is stored in a DP table. At next recurrence of the same sub-problem, the solution can be simply looked up in the table [54]. The key here is to identify useful structures in the problem (5.3) so that it can be cleanly divided into sub-problems.

Let $\Phi_{v_n}(q_{v_n}, \bar{B})$ denote the minimum distortion sum from coding unit v_n to coding unit v_N, given that v_n is coded with QP q_{v_n}, and that there is an available rate budget of $\bar{B}$, $\bar{B} \leq B$, for coding the units $v_{n+1}, \ldots, v_N$. This distortion sum $\Phi_{v_n}(q_{v_n}, \bar{B})$ can be recursively written as:

$$\Phi_{v_n}(q_{v_n}, \bar{B}) = \min_{\substack{v_{n+1} \in \mathcal{V}^o \mid v_{n+1} > v_n \\ q_{v_{n+1}} \in \mathcal{Q}}} \Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \Phi_{v_{n+1}}(q_{v_{n+1}}, \bar{B} - r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})) \quad (5.5)$$

In each recursive call, (5.5) computes the optimal unit v_{n+1} and the corresponding QP $q_{v_{n+1}}$ that minimize the distortion between v_n and v_N given the rate budget $\bar{B}$. It corresponds to minimizing the sum of the distortion between consecutive coding units v_n and v_{n+1} in $\mathcal{V}$ and the minimum distortion sum from coding unit v_{n+1} to coding unit v_N. For the latter, the budget is reduced to $\bar{B} - r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}))$ to code the units between v_{n+1} and v_N. If $v_{n+1} = V$, the recursion has reached the rightmost unit in $\mathcal{V}^o$, such that the recursive term in (5.5) is not necessary. Fig. 5.1 illustrates the structure of Eq. (5.5).

Given the first unit $v_1 = 1$ is always selected, (5.3) can be solved via the following initial call:

$$\min_{q_1 \in \mathcal{Q}} \Delta_1(q_1) + \Phi_1(q_1, B - r_1(q_1)) \quad (5.6)$$

The solution of each sub-problem $\Phi_{v_n}(q_{v_n}, \bar{B})$ is stored in entry $[v_n][q_{v_n}][\bar{B}]$ of a DP table. Hence, the complexity of the DP algorithm is bounded by the size of the DP table, VQB, multiplied by the complexity of computing each entry $O(VQ)$. Thus, the complexity of the DP algorithm is $O(V^2Q^2B)$. This is polynomial in B, but B is encoded in $\log_2(B)$ bits as input to the algorithm, and thus the algorithm is exponential in the size of the input. (This is also

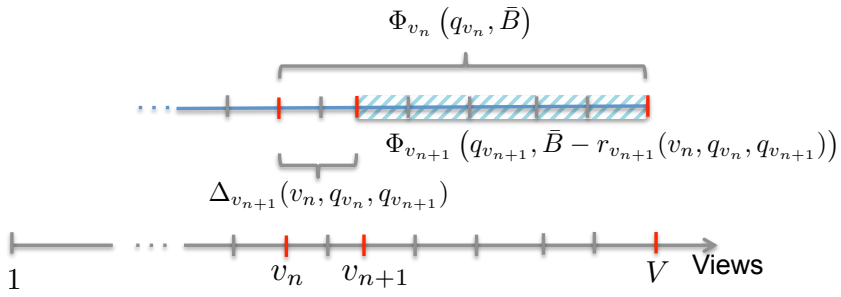


Figure 5.1 – Illustration of the distortion function $\Phi_{v_n}(q_{v_n}, \bar{B})$, which is composed by two terms. The first one is $\Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})$, that corresponds to the distortion between units v_n and v_{n+1}, coded with QPs q_{v_n} and $q_{v_{n+1}}$. The second term, $\Phi_{v_{n+1}}(q_{v_{n+1}}, \bar{B} - r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})$, corresponds to the minimum distortion sum from coding unit v_{n+1} to coding unit v_N when the budget is reduced to $\bar{B} - r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})$ to code the units between v_{n+1} and v_N.

called *pseudo-polynomial time* in the complexity literature [88]).

5.3.2 Lagrangian DP Algorithm

The DP solution of the problem in (5.5) has a relatively large complexity. We propose to reduce the complexity of the DP algorithm by eliminating the rate dimension B in the DP table. Towards that goal, we consider a Lagrange relaxed version of our constrained resource allocation problem in Eq. (5.3), where we move the rate consideration from the constraint to the objective function as a penalty term. This results in a new rate-distortion (R-D) cost formulation. The optimization problem can be rewritten as:

$$\min_{\mathcal{V},\mathbf{q}} \left[\Delta_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathcal{V}|} \Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \lambda \left(r_{v_1}(q_{v_1}) + \sum_{n=2}^{|\mathcal{V}|} r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) \right) \right] \tag{5.7}$$

where the multiplier $\lambda > 0$ is a parameter that weighs the importance of the rate against distortion in the optimal rate allocation solution.

To solve (5.7) for a given λ, we first denote $\Phi_{v_n}(q_{v_n})$ as the minimum R-D cost from the unit v_n to the unit v_N, given that v_n is coded with QP q_{v_n}. This minimum cost can be defined recursively as:

$$\Phi_{v_n}(q_{v_n}) = \min_{\substack{v_{n+1} \in \mathcal{V}^o \mid v_{n+1} > v_n \\ q_{v_{n+1}} \in \mathcal{Q}}} \Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \lambda\, r_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}}) + \Phi_{v_{n+1}}(q_{v_{n+1}}) \tag{5.8}$$

For a given λ value, in each recursive call of Eq. (5.8) the optimal unit v_{n+1} and the corre-

sponding QP $q_{v_{n+1}}$ are computed such that the distortion between v_n and v_N are minimized. The expression in Eq. (5.8) corresponds to minimizing the sum of the distortion between consecutive units v_n and v_{n+1} in $\mathcal{V}$, the λ-weighted rate of the selected unit v_{n+1}, and the minimum distortion sum from unit v_{n+1} to unit v_N. If $v_{n+1} = V$, then the recursion has reached the rightmost unit in $\mathcal{V}^o$, such that the recursive term in (5.8) is not necessary. Figure 5.2 illustrates the structure of Eq. (5.8).

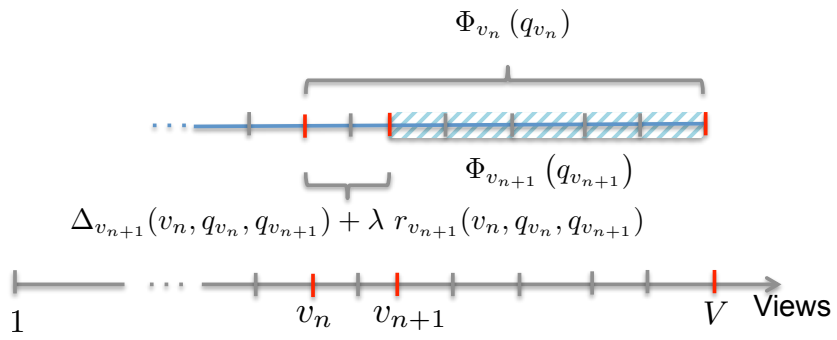


Figure 5.2 – Illustration of $\Phi_{v_n}(q_{v_n})$ definition, which is composed by three terms. The first one is $\Delta_{v_{n+1}}(v_n, q_{v_n}, q_{v_{n+1}})$, that corresponds to the distortion between units v_n and v_{n+1}, coded with QPs q_{v_n} and $q_{v_{n+1}}$. The second term stands for the λ-weighted rate of the selected unit v_{n+1}. The third term, $\Phi_{v_{n+1}}(q_{v_{n+1}})$, corresponds to the minimum distortion sum from coding unit v_{n+1} to coding unit $v_N = V$.

Given the first unit $v_1 = 1$ is always selected, (5.7) can be solved via the following initial call:

$$\min_{q_1 \in \mathcal{Q}} \Delta_1(q_1) + \lambda r_1(q_1) + \Phi_1(q_1) \tag{5.9}$$

The solution of each sub-problem $\Phi_{v_n}(q_{v_n})$ is stored in entry $[v_n][q_{v_n}]$ of a DP table. Hence, the complexity of the DP algorithm is bounded by the size of the DP table, VQ, multiplied by the complexity of computing each entry, $O(VQ)$. This results in a complexity of $O(V^2Q^2)$, which is polynomial time.

The relationship between the constrained problem in (5.3) and its Lagrangian relaxed version in (5.7) is as follows. Denote by $(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ the optimal solution of (5.7) for a given λ, solved via (5.8), with resulting distortion $D(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ and rate $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$. One can show that for a particular choice of $\lambda = \lambda^*$, if $R(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) = B$ then, $(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ is an optimal solution to (5.3) (refer to Appendix B).

However, because $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ is discrete, there may not exist a value of λ such that $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda) = B$ exactly. In this case, we can pick a value $\lambda = \lambda_1$ with an approximate Lagrangian solution $(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$, $R(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B$, with the following performance bound. Given two solutions $(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ and $(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2})$ to (5.7) using λ_1 and λ_2 with resulting rates $R(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B < R(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2})$, the difference in distortion between Lagrangian solution $(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ and the *true* optimal solution

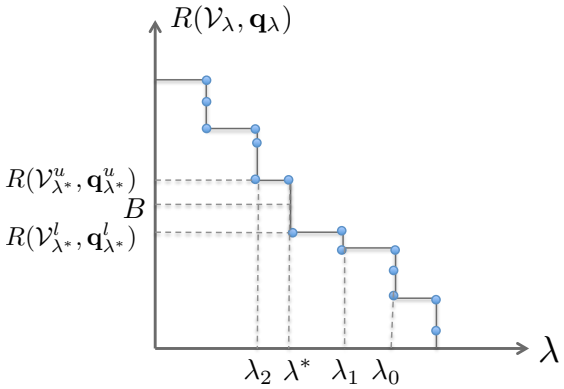


Figure 5.3 – Relationship between the rate $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ and the Lagrange multiplier λ.

$(\mathcal{V}^*, \mathbf{q}^*)$ of (5.3) is bounded:

$$|D(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathcal{V}^*, \mathbf{q}^*)| \leq |D(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2})| \tag{5.10}$$

See Appendix C for a proof. Clearly, the bound is tightest when the difference in distortion between the two Lagrangian solutions is the smallest.

We have shown how the complexity of the original DP algorithm can be reduced by our proposed Lagrangian DP solution for a given λ value. However, we still need to find the optimal λ value. Next, we propose an algorithm that efficiently finds the optimal Lagrange multiplier λ.

5.4 Optimal Lagrange Multiplier

In order to find the optimal λ value, it is important to note that $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ is a monotonically non-increasing discrete function with respect to λ. In other words, if $\lambda_1 \leq \lambda_2$, then $R(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \leq R(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$. Thus, in the search of the λ value that yields $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda) = B$, we should decrease λ if $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda) < B$ (and vice-versa). Intuitively, this means that we need to decrease the multiplier value to decrease the penalty and allow an increase of the total rate value. Moreover, as $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ is discrete, due to the discrete set of QPs $\mathcal{Q}$ considered, it implies that there are λ values at which multiple Lagrangian solutions exist; these are called *singular values* [50] [89]. Figure 5.3 illustrates the behaviour just described of the rate $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ as a function of λ. In the following, we explain the importance of these singular values of λ on the search of the optimal Lagrange multiplier and we describe how they are computed.

5.4.1 Importance of Singular Values

The notion of singular values is fundamental to find the optimal Lagrange multiplier. Two important properties of the singular values are:

1. Two neighboring singular values share one common Lagrangian solution.

2. Different values of λ between two neighboring singular values all yield to the same solution of the discrete rate allocation problem.

These two properties imply that singular values yield all solutions of the problem defined in (**??**) with λ values going from 0 to ∞. Thus, we only need to check neighboring singular values of λ in order to find the optimal one. Figure 5.3 shows sigular values of the different Lagrange multipliers, $\lambda_2, \lambda^*, \lambda_1, \cdots$, with dots representing their multiple solutions. Note that, between singular points, different λ values do not lead to new solutions.

Further, the singular value λ^* that generates the two corresponding solutions $(\mathcal{V}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*})$ and $(\mathcal{V}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$ with rate $R(\mathcal{V}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) < B < R(\mathcal{V}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$ is the best λ value that yields the best approximate Lagrangian solution $(\mathcal{V}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*})$ to (5.3). Note that the performance bound presented in Eq. (5.10) can be rewritten for this case as:

$$|D(\mathcal{V}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) - D(\mathcal{V}^*, \mathbf{q}^*)| \leq |D(\mathcal{V}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) - D(\mathcal{V}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})| \tag{5.11}$$

In the following, we propose a novel solution for the search of the singular values, given the problem defined in (5.3), which leads us to the search of the optimal Lagrange multiplier.

5.4.2 Singular Values Computation

The fact that $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ is monotone-non-increasing function with respect to λ, it means that by marching through successive singular values to span the rates of optimal rate allocation solutions $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$ to problem (5.7) towards a rate budget B, one can arrive at the best λ^* with solutions $R(\mathcal{V}^l_{\lambda^*}, \mathbf{q}^l_{\lambda^*}) < B < R(\mathcal{V}^u_{\lambda^*}, \mathbf{q}^u_{\lambda^*})$. For example, in Fig. 5.3, after testing λ_0 and λ_1 successively, we arrive at the best value λ^* that satisfy the rate constraint B with the performance bound given in (5.11). The technical challenge thus consists in marching through successive singular values efficiently towards the optimal λ^*.

Towards this goal, we first define $(v^*_{n+1}, q^*_{v_{n+1}})$ to be the argument that minimizes the sub-problem $\Phi_{v_n}(q_{v_n})$ in (5.8) for a given λ. We further denote by $\Psi_{v_n}(q_{v_n})$ and $\Upsilon_{v_n}(q_{v_n})$ the distortion and the rate of the sub-problem $\Phi_{v_n}(q_{v_n})$, which can be computed using the solution $(v^*_{n+1}, q^*_{v_{n+1}})$ and stored in similar DP tables when (5.8) is solved recursively. We have:

$$\Psi_{v_n}(q_{v_n}) = \Delta_{v^*_{n+1}}(v_n, q_{v_n}, q_{v^*_{n+1}}) + \Psi_{v^*_{n+1}}(q_{v^*_{n+1}}) \tag{5.12}$$

$$\Upsilon_{v_n}(q_{v_n}) = r_{v^*_{n+1}}(v_n, q_{v_n}, q_{v^*_{n+1}}) + \Upsilon_{v^*_{n+1}}(q_{v^*_{n+1}}) \tag{5.13}$$

To find the neighboring singular value λ^-, where $\lambda^- < \lambda$, we know that λ^- and λ share an optimal solution, and that λ^- has an additional solution with rate larger than the rate of the shared solution. This additional global solution $(\mathcal{V}_{\lambda^-}, \mathbf{q}_{\lambda^-})$ may stem from a new solution of the sub-problem $\Phi_{v_n}(q_{v_n})$ as λ decreases. In particular, for each sub-problem $\Phi_{v_n}(q_{v_n})$ we find the neighboring singular value $\lambda^-_{v_n}(q_{v_n})$, where $\lambda^-_{v_n}(q_{v_n}) < \lambda$, considering only the entry $[v_n][q_{v_n}]$ in the DP-table.

First, let $\Upsilon_{v_n}(q_{v_n})$, as defined in (5.13), be the common optimal solution of $\lambda^-_{v_n}(q_{v_n})$ and λ. Since $\lambda^-_{v_n}(q_{v_n})$ is singular, we know that there is at least one more solution in addition to $\Upsilon_{v_n}(q_{v_n})$. Thus, there exists a $v \in \mathcal{V}^o | v > v_n$ and a $q_v \in \mathcal{Q}$ such that:

$$\Psi_{v_n}(q_{v_n}) + \lambda^-_{v_n}(q_{v_n})\Upsilon_{v_n}(q_{v_n}) = (\Delta_v(v_n, q_{v_n}, q_v) + \Psi_v(q_v)) + \lambda^-_{v_n}(q_{v_n})(r_v(v_n, q_{v_n}, q_v) + \Upsilon_v(q_v)) \tag{5.14}$$

For other values of v and q_v, we have:

$$\Psi_{v_n}(q_{v_n}) + \lambda^-_{v_n}(q_{v_n})\Upsilon_{v_n}(q_{v_n}) \leq (\Delta_v(v_n, q_{v_n}, q_v) + \Psi_v(q_v)) + \lambda^-_{v_n}(q_{v_n})(r_v(v_n, q_{v_n}, q_v) + \Upsilon_v(q_v)) \tag{5.15}$$

Then, since $\lambda^-_{v_n}(q_{v_n})$ is the closest singular value from below to λ, it is computed as:

$$\lambda^-_{v_n}(q_{v_n}) = \max_{\substack{v \in \mathcal{V}^o | v > v_n \\ q_v \in \mathcal{Q}}} \frac{\Psi_{v_n}(q_{v_n}) - (\Delta_v(v_n, q_{v_n}, q_v) + \Psi_v(q_v))}{(r_v(v_n, q_{v_n}, q_v) + \Upsilon_v(q_v)) - \Upsilon_{v_n}(q_{v_n})} \tag{5.16}$$

where the search for the maximization is over the set of units v and QPs q_v with a resulting rate $r_v(v_n, q_{v_n}, q_v) + \Upsilon_v(q_v) > \Upsilon_{v_n}(q_{v_n})$. In other words, $\lambda^-_{v_n}(q_{v_n})$ is the closest λ value at which sub-problem $\Phi_{v_n}(q_{v_n})$ will result in a different solution as λ decreases.

Note that the birth of a new global solution $(\mathcal{V}_{\lambda^-}, \mathbf{q}_{\lambda^-})$ can stem from any sub-problem $\Phi_{v_n}(q_{v_n})$

as λ decreases. Thus λ^- is the largest of all $\lambda^-_{v_n}(q_{v_n})$:

$$\lambda^- = \max_{v_n \in \mathcal{V}^o, q_{v_n} \in \mathcal{Q}} \lambda^-_{v_n}(q_{v_n}) \tag{5.17}$$

Similarly, the neighboring singular value λ^+, where $\lambda^+ > \lambda$, is computed for each sub-problem $\Phi_{v_n}(q_{v_n})$ as:

$$\lambda^+_{v_n}(q_{v_n}) = \min_{\substack{v \in \mathcal{V}^o | v > v_n \\ q_v \in \mathcal{Q}}} \frac{\left(\Delta_v(v_n, q_{v_n}, q_v) + \Psi_v(q_v)\right) - \Psi_{v_n}(q_{v_n})}{\Upsilon_{v_n}(q_{v_n}) - (r_v(v_n, q_{v_n}, q_v) + \Upsilon_v(q_v))} \tag{5.18}$$

Then, the singular value λ^+ is the smallest of all $\lambda^+_{v_n}(q_{v_n})$:

$$\lambda^+ = \min_{v_n \in \mathcal{V}^o, q_{v_n} \in \mathcal{Q}} \lambda^+_{v_n}(q_{v_n}) \tag{5.19}$$

Updating the DP table for the new Lagrange multiplier, λ^- or λ^+, has a complexity of $\mathcal{O}(VQ)$, and not of $\mathcal{O}(V^2Q^2)$ that refers to the complexity of creating the DP table for an initial λ. The complexity is reduced since it only depends on the size of the DP table as the update computation of each entry is constant.

We have presented in this section a method to update λ towards its optimal value λ^*. Next, we define a good starting value for λ.

5.4.3 Initial Lagrange Multiplier Value

The complexity of the algorithm for searching through the singular values depends on the initial guess of a good Lagrange multiplier value. The closer this initial guess is to the optimal value, the fewer the number of iterations of the above search algorithm until reaching λ^*. We start by defining a possible optimal solution $\{\hat{\mathcal{V}}, \hat{\mathbf{q}}\}$ for a given λ, *e.g.*, a selection of N equally spaced units that are all coded with the same QP q, which corresponds to an average of the QPs available in $\mathcal{Q}$. This has been experimentally proven to be close to optimal solution. Then, we consider a different case where the QP for a unit $v_n \in \mathcal{V}^*$ is $q_{v_n} = q + a$, while the rest of the units are all coded using the same QP q. If $\{\hat{\mathcal{V}}, \hat{\mathbf{q}}\}$ is a solution of (5.7) for a given λ, then the

following condition is met:

$$\Delta_{v_{n+1}}(v_n, q, q) - \Delta_{v_{n+1}}(v_n, (q+a), q) \leq \lambda \left(r_{v_{n+1}}(v_n, q, q) - r_{v_{n+1}}(v_n, q+a, q)\right) \tag{5.20}$$

Then, the initial value of λ can be approximated as :

$$\lambda \approx \frac{\Delta_{v_{n+1}}(v_n, q, q) - (\Delta_{v_{n+1}}(v_n, (q+a), q)}{r_{v_{n+1}}(v_n, q, q) - r_{v_{n+1}}(v_n, q+a, q)} \tag{5.21}$$

Note that a different initialization can be also used with our algorithm.

5.4.4 Lagrange Multiplier Search Algorithm

Given the definitions above, we now have an algorithm to solve the constrained problem in (5.3) through an unconstrained formulation of the problem using Lagrangian cost function. The algorithm is based on an optimized search of the Lagrange multiplier using the concept of singular values, through Eq. (5.16) and Eq. (5.18). More formally, the search strategy for the best multiplier λ^* to obtain the closest approximate Lagrangian solution to (5.3) can be described as follows:

Step 1 Initalize λ (described in Section 5.4.3) and solve (5.7) via algorithm (5.8) with unique solution $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda)$.

Step 2 If $R(\mathcal{V}_\lambda, \mathbf{q}_\lambda) < B$, find next smaller singular value λ^- via (5.17), $\lambda^- < \lambda$. Otherwise, find next larger singular value λ^+ via (5.19), $\lambda^+ > \lambda$.

Step 3 Find simultaneous solutions $(\mathcal{V}_\lambda^l, \mathbf{q}_\lambda^l)$ and $(\mathcal{V}_\lambda^u, \mathbf{q}_\lambda^u)$ for the selected singular value, where $R(\mathcal{V}_\lambda^l, \mathbf{q}_\lambda^l) < R(\mathcal{V}_\lambda^u, \mathbf{q}_\lambda^u)$.

Step 4 If $R(\mathcal{V}_\lambda^u, \mathbf{q}_\lambda^u) < \bar{B}$, find next smaller singular value λ^- via (5.17), $\lambda^- < \lambda$. Goto step 3.

Step 5 If $\bar{B} < R(\mathcal{V}_\lambda^l, \mathbf{q}_\lambda^l)$, find next larger singular value λ^+ via (5.19), $\lambda^+ > \lambda$. Goto step 3.

Step 6 If $R(\mathcal{V}_\lambda^l, \mathbf{q}_\lambda^l) \leq \bar{B} \leq R(\mathcal{V}_\lambda^u, \mathbf{q}_\lambda^u)$, declare $(\mathcal{V}_\lambda^l, \mathbf{q}_\lambda^l)$ as the best approximate Lagrangian solution. Stop.

5.5 Performance Assessment

We now evaluate the performance of our new optimized Lagrange multiplier selection in rate allocation problems. In addition to multiview video settings, we also consider traditional

monoview video cases. Indeed, our formulation and our new rate allocation algorithm are general enough to encompass both cases, meaning that coding units can be either views in a multiview video scenario or frames in a monoview video.

We considered the monoview video datasets *Hall Monitor* (352 × 288, 30fps)[90] [91] and *Kimono* (1920 × 1080, 24fps), provided by Tokyo Institute of Technology - Nakajima Laboratory; both sequences have a GOP size of 1s, with 30 frames and 24 frames, respectively. For the multiview video datasets, we used three sequences: *Shark* (1920 × 1088, 30 fps, 9 views), provided by NICT for MPEG FTV standardization [1], *Undo Dancer* (1920 × 1088, 25 fps, 5 views) [79] and *Soccer Linear2* (1600 × 1200, 60 fps, 7 views) [92]. The camera views are equally spaced for the *Shark* and the *Soccer Linear2* datasets, with inter-view separation of $70.04mm$ and $2m$, respectively. For *Undo Dancer* sequence, only the camera views $\{1,2,3,5,9\}$ are available in the dataset. Thus, these views are possibly chosen for coding in our algorithm and the remaining views $\{4,6,7,8\}$ can be synthesized at the decoder. Finally, the interview separation distance in the set of available views at the user is of $20cm$. We used a GOP size of 8 frames and an intra-period of 24 frames as defined under the common test conditions by JCT-3V [93]. In Fig. 5.4 a frame of each of the considered monoview and multiview video is illustrated.

Our algorithm finds only one optimal QP value for each unit selected for coding. Thus, in the case of traditional monoview video, the set of frames for coding and their corresponding QP are optimized with our rate allocation algorithm. In the case of multiview video, we optimize the set of views to encode and the QP of each view. As each view has a texture and depth components and each one has a set of frames, instead of coding all the frames in a view with the same selected QP value, we follow the following strategy:

1. We assume that the depth maps are all encoded at the same high quality (QP=30), as accurate depth information is important for view synthesis. Note that a different QP value could have been used to encode the depth maps, we have chosen QP=30 as it offers a good compromise between view synthesis quality and rate for considered sequences. Thus, in our rate allocation algorithm we only optimize the QP of the texture information.

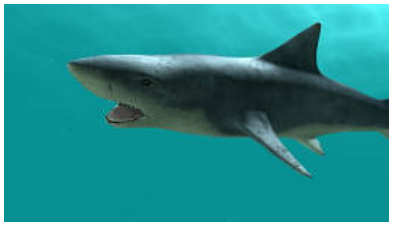
2. We adopt the hierarchical B-frames/slices coding mode [85] in the temporal domain of each view, where B-frames are hierarchically predicted from other B or anchor frames (i.e. frames that do not have any temporal prediction), where a cascading quantization parameters (CQP) [65] strategy is normally used, as suggested in the reference software of 3D-HEVC. This means that by finding the optimal QP value for the texture of each view, we only find the optimal QP value for the anchor frames and the CQP strategy is used for the rest of frames in the GOP. In this strategy, the anchor frames of the texture and depth are encoded with the optimized QP value and QP=30, respectively; while the other frames in the GOP use a ΔQP that is added to the selected QP value of the anchor frames. This ΔQP depends on the position of the frame in the GOP, where references frames

(a) *Hallmonitor* monoview sequence, frame 30.

(b) *Kimono* monoview sequence, frame 150.

(c) *Soccer Linear2* multiview sequence, view 1, frame 1.

(d) *Undo Dancer* multiview sequence, view 3, frame 1.



(e) *Shark* multiview sequence, view 20, frame 1.

Figure 5.4 – Content characteristics example for a frame of each considered monoview or multiview sequence: (a) *Hallmonitor*, (b) *Kimono*, (c) *Soccer Linear2*, (d) *Undo Dancer*, (e) *Shark*.

used for the prediction of frames in higher temporal layers have a lower ΔQP. Given the 8 frames of a GOP for the considered dataset, we used a ΔQP= $\{0,1,2,3,4,4,3,4\}$ as suggested in the reference software of 3D-HEVC.

In our experiments, we compute the distortion in terms of PSNR of the luminance (Y-PSNR) that is evaluated on both coded and synthesized units, in case units are dropped at the encoder. To reconstruct missing frames in monoview videos we considered a commonly used method based on motion estimation [94][95], where given two frames v_0 and v_1 and an estimate of

the optical flow f between these frames, a missing frame v_i, $i \in (0,1)$ is estimated as:

$$v_i(x) = 1/2(v_0(x+af) + v_1(x-(1-a)f)), \tag{5.22}$$

where a represents the distance between the reference frames v_0, v_1 and the reconstructed frame. For the missing views in multiview video, we used a simple depth-image based rendering (DIBR) method at the decoder where pixels from the closest right and left coded views are projected to the intermediate missing viewpoint using the texture information and the intensity of the depth value per pixel. Then, the projected pixels from the reference views are merged together, e.g., using a linear weighting function that considers the distance between reference and virtual views [96].

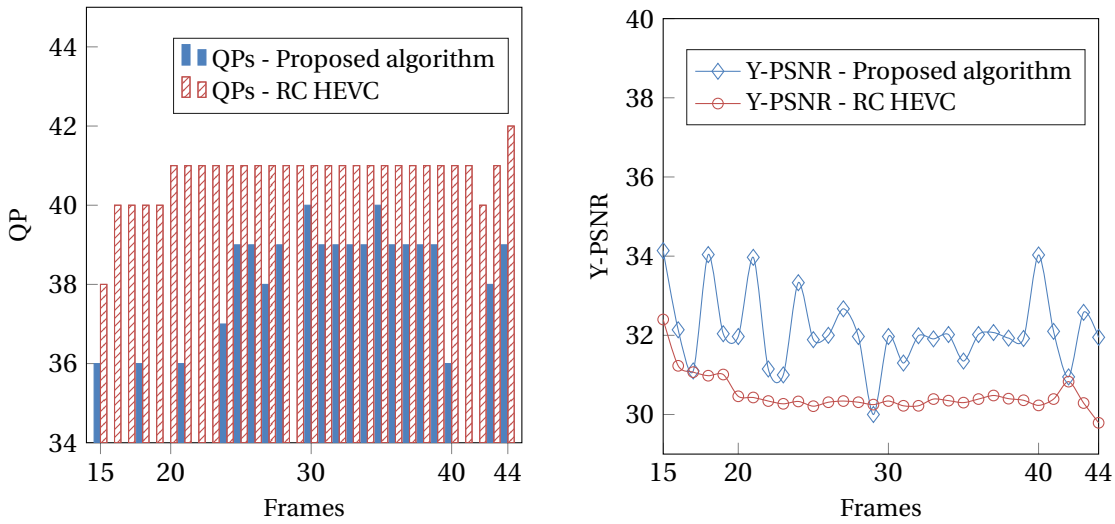
To better illustrate the performance of our algorithm, we also show the performance of the rate control (RC) solutions [9][10] adopted by the reference software HM 15.0 [97] of the High Efficiency Video Coding (HEVC) standard [85], for monoview videos, and by the reference software HTM 13.0 [98] of the 3D extension of HEVC (3D-HEVC) [99], for multiview video. These solutions only optimize the QPs of the different frames and they do not skip units at the encoder. For a more fair illustration of both solutions, in the multiview video case, we also also fix the QP value of the depth maps when the rate control of the reference software for 3D-HEVC is tested, so only the QPs of the texture component of the views are optimized.

In the following, the performance of our algorithm is evaluated for monoview and multiview videos considering two scenarios: (i) when units are independently coded and (ii) when units are predictively coded.

5.5.1 Experiments with Independently Coded Units

We first evaluate the performance of our rate allocation algorithm in the case of independently encoded units for monoview and multiview video sequences. The available set of QPs for the coding units are $\mathcal{Q} = \{25, 26, \cdots, 51\}$ for both cases.

In Fig. 5.5a, the QP selection for each frame of our algorithm solution is compared to the QP adaptation solution of the RC of HEVC, for one GOP (from frame 15 to 44) of the *Hall Monitor* monoview sequence. For these results, we consider a rate budget of 500 kbps where our algorithm achieved a rate of 490.60 kbps and the RC of HEVC a rate of 499.53 kbps. Note that a QP=0 means that the frame is skipped and needs to be reconstructed at the decoder. Most of the frames that are skipped with our algorithm are the frames between frames 15 and 23, which corresponds to the lowest motion part in the GOP under consideration. In addition, it can be seen that our algorithm assigns low QP values to the frames that are neighbors of the dropped frames, since they are used as reference frames in their reconstruction at the decoder. In Fig. 5.5b, we show the quality values (Y-PSNR) of coded (QP > 0) and reconstructed (QP



(a) (b)

Figure 5.5 – Frame-to-frame comparison of our proposed algorithm and the RC of HEVC for *Hall Monitor* monoview video sequence: (a) QP selection and (b) quality comparison (Y-PSNR). Rate budget $B = 500kbps$, with our proposed solution rate $R = 490.60kbps$, and the rate of RC of HEVC rate $R = 499.53kbps$.

=0) frames, for the same test as in Fig. 5.5a. It can be seen how the solution of the proposed algorithm achieves a higher average quality compared to the solution of the RC of HEVC, achieving an average Y-PSNR=32.11 dB, while the RC solution has an average Y-PSNR=30.50 dB. The quality fluctuations are due to the drop in quality when frames are skipped at the encoder and reconstructed at the decoder. The proposed algorithm takes advantage of the frame skipping property to skip frames when the motion is low in the sequence, so that it save bits to enhance the quality of coded frames in counterpart.

The QP selection for each frame in each view in the multiview video case is illustrated in Fig. 5.6 for our algorithm and for the RC of 3D-HEVC. We consider the *Shark* multiview video sequence and a rate budget of 250 kbps. In this solution, the consumed rate of the proposed algorithm is 245.49 kbps, while the RC of 3D-HEVC uses 242.55 kbps. We observe that no view is skipped by our algorithm. For the same conditions a view to view quality comparison, resulting from the QP selection shown in Fig. 5.6, is now illustrated in Fig. 5.7. In general, our algorithm achieves a higher average Y-PSNR, 30.49 dB, compared to the RC of 3D-HEVC, 29.98 dB. Given the specified rate budget, the RC of 3D-HEVC solution selects a low QP to encode the anchor frame of each view and the other frames in the GOP are encoded using the maximum QP value (QP=51). Differently, by adopting the CQP strategy, the QP selected for the anchor frame by our algorithm is always higher than the ones used by the RC of 3D-HEVC, but the QPs of the inter-predicted frames can be lower. By using this strategy, our algorithm performs better in terms of average quality.

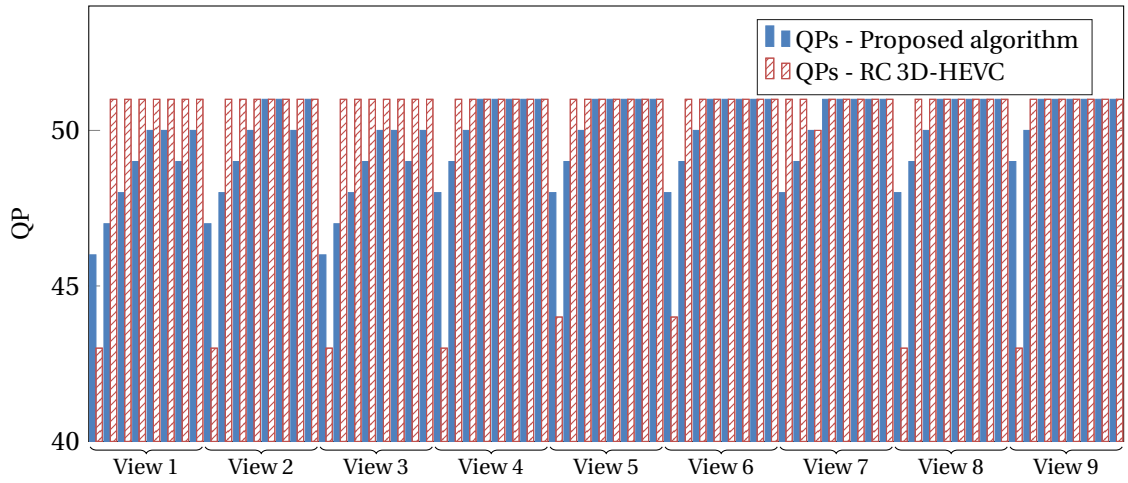


Figure 5.6 – Frame-to-frame per view comparison of the QP selection of our proposed algorithm and the RC of 3D-HEVC for *Shark* multiview video sequence. Rate budget $B = 250kbps$, with our proposed solution rate $R = 245.49kbps$, and the rate of RC of 3D-HEVC $R = 242.55kbps$.

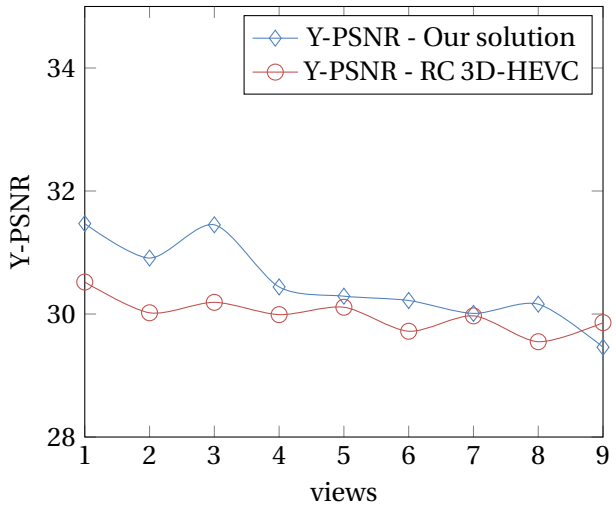


Figure 5.7 – View-to-view average quality comparison (Y-PSNR) of our proposed algorithm and the RC of 3D-HEVC for *Shark* multiview video sequence. Rate budget $B = 250kbps$, with our proposed solution rate $R = 245.49kbps$, and $R = 242.55kbps$ in RC of 3D-HEVC.

Tables 5.1 and 5.2 present the performance of both, our rate allocation algorithm and the RC of HEVC, in terms of average Y-PSNR given a rate budget B for the monoview sequences *Hall Monitor* and *Kimono*. The difference, ΔY-PSNR, between both quality values is also presented where a positive value means a quality gain of the proposed algorithm. From the results, we can see that our algorithm always gets a solution with a rate that is under the rate budget B and it always achieves the highest quality, with a Δ Y-PSNR of up-to $2.34dB$. The visual quality is illustrated in Fig. 5.8 for *Hall Monitor* for frames 15 and 17, when the rate budget

is B = 150 kbps. Our algorithm tends to skip frames with low motion, as frame 17, which are then reconstructed at the decoder achieving a final higher visual quality compared to the RC of HEVC that uses a higher QP value to satisfy the rate budget.

Table 5.1 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, given the *Hall Monitor* monoview video sequence with independently encoded frames.

B	Proposed algorithm		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
150	149.39	26.62	149.96	24.93	1.69
200	198.13	27.60	200.16	25.26	2.34
300	298.88	29.57	300.31	27.29	2.28
400	366.10	30.66	400.09	28.98	1.68
500	490.60	32.11	499.53	30.50	1.61

Table 5.2 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, given the *Kimono* monoview video sequence with independently encoded frames.

B	Proposed algorithm		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
150	149.03	27.44	149.81	26.42	1.02
200	198.80	28.59	200.46	27.20	1.39
300	296.32	29.56	299.97	28.37	1.19
400	391.20	30.03	400.36	29.35	0.68
500	499.61	30.23	502.06	30.17	0.06

Similarly, Tables 5.3, 5.4 and 5.5 present the performance of our algorithm and the RC of 3D-HEVC in terms of average Y-PSNR and ΔY-PSNR, with given rate budgets for the multiview sequences *Shark*, *Undo Dancer* and *Soccer Linear2*. In general, the gains in quality are smaller for multiview video sequences compared with the monoview sequences. The main reason is that by skipping frames in traditional monoview videos, a higher average quality could be achieved. However, multiview video views are not easily skipped as the impact on the final average quality is larger. In addition, we only optimize one QP per view, the QP of the anchor frame of each view while the CQP strategy is used for the other frames in the GOP. Differently, the RC of 3D-HEVC adapts the QP of all the frames in each view, getting rate values that are closer to the rate budget. This is the case of the results for *Soccer Linear2* sequence with a rate budget of 250 kbps and 400 kbps, where our solution has a slightly lower quality compared to the solutions of the RC of 3D-HEVC.

(a) Frame 15 - Proposed algorithm - QP= 43

(b) Frame 15 - RC HEVC - QP= 47

(c) Frame 17 - Proposed algorithm - Reconstructed frame

(d) Frame 17 - RC HEVC - QP= 51

Figure 5.8 – Visual quality illustration for the *Hall Monitor* monoview video sequence with independently encoded frames when the proposed algorithm and the RC of HEVC are used (B = 150 kbps). (a) and (b) Show frame 15 encoded according to our proposed algorithm and the RC of HEVC, respectively. (c) Shows frame 17, that has has been skipped at the encoder and reconstructed at the decoder according to the proposed algorithm, achieving a higher visual quality compared to the RC of HEVC output in (d).

5.5.2 Experiments with Predictively Coded Units

We consider now the predictive coding case, in particular, an $IPP\cdots$ coding model for both monoview (inter-frame coding model) and multiview video (inter-view coding model). As the computational complexity due to coding increases, compared to the independently coded case, we decrease the granularity of the available QPs in our search space to $\mathcal{Q} = \{25, 28, 31\cdots, 51\}$

For different rate constraints B, Tables 5.6 and 5.7 show the performance, in terms of rate and average quality, of our proposed algorithm and the RC of HEVC for the *Hall Monitor* and *Kimono* monoview sequences. As mentioned in Section 5.2.1, due to complexity reasons, in

Table 5.3 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of 3D-HEVC, given the *Shark* multiview video sequence with independently encoded views.

B	Proposed algorithm		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
175	174.41	29.25	162.45	28.22	1.03
200	196.26	29.26	198.75	29.22	0.04
300	287.61	31.08	287.49	30.57	0.51
400	381.54	32.35	389.28	31.51	0.84
500	481.71	33.49	485.13	32.22	1.27

Table 5.4 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of 3D-HEVC, given the *Undo Dancer* multiview video sequence with independently encoded views.

B	Proposed algorithm		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
175	169.15	27.71	169.55	26.99	0.72
200	199.15	28.39	197.32	27.51	0.88
300	280.55	29.57	286.95	28.74	0.83
400	383.05	30.79	380.70	29.69	1.1
500	476.78	31.63	486.15	30.47	1.16

Table 5.5 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of 3D-HEVC, given the *Soccer Linear2* multiview video sequence with independently encoded views.

B	Proposed algorithm		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
175	172.90	28.15	173.28	27.70	0.45
250	245.98	28.89	249.06	28.99	-0.1
300	296.23	29.44	288.82	29.24	0.2
400	395.32	30.30	398.11	30.77	-0.47
500	487.4	31.42	482.24	31.03	0.39

this work, we assume that each coded unit only depends on the previous coded one and not on all previous coded units, until the first independently coded unit. However, this assumption tends to underestimate the coding rate, as it limits the effect of previously encoded units to

the first reference view used for prediction. Thus, to solve this problem, we adopt a post-processing method, where previous solutions of our algorithm (*i.e.*, when marching towards the optimal Lagrange multiplier, moving from a lower to a higher rate, thus using Eq. (5.18)) are saved and the first one satisfying the rate constraint is selected as the new solution. For this reason, in Tables 5.6 and 5.7 the results of the proposed algorithm is showed in two columns, (I) and (II). In the first column, the solution of our original algorithm is presented, where the rate is usually slightly higher than the rate budget. In the second column, the adjusted values obtained with the post-processing step just described are shown. This second column repeats first column result if the original algorithm already satisfies the rate constraint B. Compared to the independent coding case, in the predictive coding scenario, the RC of HEVC did not show a good performance for the two monoview video sequences considered. This is evident for the *Kimono* dataset where the rate of some solutions are far above from the rate budget constraint, in particular for low rate constraint values. Thus, it becomes difficult to compare both solutions performance in terms of quality. However, it can be seen that, in the cases where the RC of HEVC achieves a good rate value (*i.e.*, under or close to the rate constraint), our algorithm achieves a better average quality or close to HEVC performance. Moreover, we use a coarser set of QPs than the RC of HEVC, so that our results could be generally improved if the same set of QPs $\mathcal{Q}$ is used by both schemes. Compared to the independently coded units case, the gain achieved by our algorithm is now smaller. This is due to the fact that when frames are predictively encoded, skipped frames have higher impact in the overall quality as the distance increases between a coded frame and its reference used for prediction.

Table 5.6 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm (first and second solution) and for the RC of HEVC, given the *Hall Monitor* monoview video sequence with predictively coded frames.

B [$kbps$]	Proposed algorithm (I)	Proposed algorithm (II)	RC HEVC	Δ Y-PSNR [dB]
	R [$kbps$]	R [$kbps$]	R [$kbps$]	
	Y-PSNR [dB]	Y-PSNR [dB]	Y-PSNR [dB]	
25	28.94	24.41	37.49	
	27.92	27.06	28.10	-1.04
50	57.62	47.96	50.14	
	30.46	30.06	29.92	0.14
75	78.77	74.82	75.06	
	34.35	33.92	33.70	0.22
100	105.06	96.84	99.89	
	34.69	34.17	34.28	-0.11
150	156.34	148.16	150.09	
	36.66	36.26	36.02	0.24

Table 5.7 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm (first and second solution) and for the RC of HEVC, given the *Kimono* monoview video sequence with predictively coded frames.

B [$kbps$]	Proposed algorithm (I)	Proposed algorithm (II)	RC HEVC	Δ Y-PSNR [dB]
	R [$kbps$]	R [$kbps$]	R [$kbps$]	
	Y-PSNR [dB]	Y-PSNR [dB]	Y-PSNR [dB]	
50	59.88	48.80	79.60	
	26.88	26.33	27.09	-0.76
75	80.94	73.01	96.44	
	27.70	27.21	28.01	-0.8
100	113.99	97.06	120	
	28.96	28.40	29.02	-0.62
150	155.31	148.34	152.44	
	30.96	29.75	29.54	0.21
200	186.44	186.44	204.12	
	30.41	30.41	30.24	0.17

Finally, Tables 5.8, 5.9 and 5.10 present the performance in terms of rate and average quality of our proposed algorithm and the RC of 3D-HEVC for the *Shark*, *Undo Dancer* and *Soccer Linear2* multiview video sequences. Here, compared to the monoview video cases in Tables 5.6 and 5.7, the proposed algorithm has a better performance, as the obtained solution tends to satisfy the rate budget most of the time with our original algorithm and there is usually not need to use the post-processing step. This is due to the length in the prediction paths. In the case of multiview video, the maximum length of the (inter-view) prediction path is 9 views (e.g., *Shark*), compared to 30 and 24 frames (GOP size) for *Hall Monitor* and *Kimono* monoview video sequences. This means that, for the multiview video case, the effect of previously coded units in a current predicted unit is much more limited than in monoview video cases, thus making our assumption more reasonable. In general, from these results we can conclude that when our algorithm is close to the rate budget (*i.e.*, the granularity of the available QPs is not affecting the solution) it achieves a higher overall quality than the RC of 3D-HEVC.

5.6 Conclusions

A new solution for the optimal selection of the Lagrange multiplier in Lagrangian-based rate allocation optimization problems has been addressed in this chapter. We have considered a general rate allocation formulation that can be applied in different scenarios, in particular for multiview video with views that are independently or predictively coded. Given the high

Table 5.8 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, given the *Shark* multiview video sequence with predictively coded views.

B	Our solution		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
75	74.48	28.97	107.64	28.03	0.94
100	98.19	29.15	107.64	28.03	1.12
150	147.25	31.02	147.60	29.51	1.51
200	195.47	32.68	190.59	30.69	1.99
300	297.22	33.51	290.91	32.59	0.92

Table 5.9 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm and for the RC of HEVC, given the *Undo Dancer* multiview video sequence with predictively coded views.

B	Our solution		RC HEVC		
$[kbps]$	R $[kbps]$	Y-PSNR $[dB]$	R $[kbps]$	Y-PSNR $[dB]$	Δ Y-PSNR $[dB]$
50	49.93	25.87	46.85	25.32	0.55
75	74.15	28.23	74.75	27.54	0.69
100	76.72	28.29	82.22	27.96	0.33
150	129.81	30.25	148.07	30.47	0.22
200	191.9	32.12	194.00	31.50	0.62

complexity of a classic dynamic programming (DP) algorithm, we modified the problem formulation to consider a Lagrangian cost function that permits to reduce the complexity of the DP solution. Moreover, we proposed a new method to optimally select the Lagrange multiplier in these types of problems. Rate control solutions adopted by the reference softwares of current monoview and multiview video reference encoders, HEVC and 3D-HEVC, have been used to illustrate the performance of our proposed algorithm. Overall, our proposed simple solution compares favorably to these complex solutions in terms of average Y-PSNR when frames (in monoview video) and views (in multiview video) are independently or predictively coded.

Table 5.10 – Rate budget B, actual rate R and average Y-PSNR value for the proposed algorithm (first and second solution) and for the RC of HEVC, given the *Soccer Linear2* multiview video sequence with predictively coded views.

B [$kbps$]	Proposed algorithm (I)	Proposed algorithm (II)	RC HEVC	Δ Y-PSNR [dB]
	R [$kbps$]	R [$kbps$]	R [$kbps$]	
	Y-PSNR [dB]	Y-PSNR [dB]	Y-PSNR [dB]	
100	98.96	98.96	93.06	
	28.06	28.06	26.09	1.97
200	180.48	180.48	184.08	
	30.77	30.77	29.72	1.05
250	252.70	180.48	247.02	
	32.68	30.77	31.04	-0.27
300	303.72	296.27	289.02	
	33.33	33.27	32.18	1.09
400	396.18	396.18	376.14	
	34.71	34.71	33.53	1.18

6

Conclusions

6.1 Main Contributions

Application specific coding strategies for IMV is key to ensure mass acceptance by users of novel interactive services in the near future. The role of efficient coding strategies becomes much more important in multiview applications, due to the enormous amount of data that needs to be stored and transmitted. This has been the main target of this thesis, where, we have described current limitations in IMV and we have proposed novel coding solutions to provide high quality interactive services in resource constrained settings.

First, in the scenario of IMVS, where users periodically request view switches, we have proposed a greedy algorithm to find the optimal interview PS and QPs for the texture and depth maps given an MVD representation of the data, such that the amount of data transmitted to the user is minimized. The challenge here is that, without knowing the path that each user will follow in his/her interactive navigation, the content should be efficiently coded a priori. Differently from most of the works in the literature that target efficient compression of the multiview data where all the views are stored and transmitted together, the optimal PS and QPs resulting from our algorithm trades-off transmission rate or interaction flexibility and compression efficiency. In particular, the optimal PS and QPs minimize the distortion in a system where the point-to-point transmission bandwidth and the storage capacity are scarce resources. We have shown through simulations that the proposed algorithm is able to identify a near-optimal PS in the sense of minimizing the distortion while trading off the transmission and storage costs. Moreover, compared with an exhaustive search approach the associated complexity is considerably reduced.

Next, we have considered the scenario where access link bandwidth capabilities can be heterogeneous across the network. In this context, we proposed a *layered multiview representation*, a solution that adapts to the different bandwidth capabilities of the clients. In this solution, the users are clustered according to their bandwidth capabilities and the set of views transmitted to each group of clients, and their corresponding rate, are carefully selected so that their navigation quality is maximized. The layered organization of the views allows a scalable solution, where users with low bandwidth capabilities have access to only the first layers and users with higher capabilities are able to receive more layers (more views), hence benefiting of a better navigation quality. We proposed a globally optimal solution based on the dynamic-programing (DP) algorithm and a greedy reduced-complexity algorithm, where the optimal subset of views and their coding rates are computed successively for each layer by a DP-based approach. Simulation results have shown that our greedy algorithm achieves a close-to-optimal performance in terms of total expected distortion, and outperforms a distance-based view and uniform rate allocation strategy used as a baseline algorithm for the layer construction.

Finally, we have focused on the rate allocation problem, where the goal is to optimally distribute a rate budget among a set of views (named, *coding units*) in a multiview video. Based on the greedy algorithm presented in the context of adaptive solutions for IMV, we have proposed a rate allocation algorithm that combines dynamic programming and Lagrange optimization to further decrease the complexity of the rate allocation algorithm. The main contribution resides in effectively finding the optimal Lagrange multiplier. To study the performance of our proposed algorithm, we have considered both multiview video and traditional monoview video demonstrating the generalization capabilities of the algorithm. Moreover, to appreciate the results of our algorithm we used the rate control solutions adopted in the reference softwares of current monoview and multiview video standards, namely HEVC and 3D-HEVC. We showed how our solution compares favorably with these two more complex rate control methods.

Overall, in this thesis we have proposed different coding strategies for solving problems arising in the context of IMV. In particular, the main contributions of this thesis can be summarized as: (i) an algorithm that permits to efficiently use the bandwidth available and storage capacity (*i.e.*, trading off the interaction flexibility and the compression efficiency) by optimizing inter-view dependencies on the PS, where the user preferences are considered. (ii) an adaptive solution for IMV service in heterogeneous networks, and (iii) an algorithm for optimal Lagrange multiplier for Lagrangian optimization in rate allocation problems for multiview video.

6.2 Future Directions

While this thesis has demonstrated the importance of effective coding solutions for IMV, many opportunities for extending the scope of this thesis remain. This section presents some of

these future research directions.

In Chapter 3, we have proposed an algorithm that selects the optimal PS and QPs of texture and depth maps for IMVS. However, a unique QP is found for a given PS (one for the texture and one for the depth map), under the assumption that different views in the same dataset are very similar. An interesting extension of this work would be to optimize the QPs of the different views considering the popularity distribution across the views, where views that are more popular among the users have higher quality (low QP value) and vice-versa. In addition, a future research may focus on the extension of the current optimization algorithm to systems where the reference views used to synthesize the considered virtual views are not restricted to be the closest ones, but their choice can be optimized to further improve the performance of our algorithm. Also, the implementation of a non-static view temporal popularity model is left for future work, where frames popularity change over the time.

When optimizing the PSs of texture and depth maps for IMVS an important component that should be consider is the energy consumption. For instance, in embedded systems the memories are limited by power constraint and encoding/decoding and view synthesis in multiview video are processes that pose the need of drastically reducing the "memory bandwidth" consumption. The selection of PSs are a dominant factor in energy efficient multiview video systems. The energy consumption is related to the type of frames used, where I-frames are the lightest ones as motion and disparity estimation (ME/DE) are skipped and B-frames require the highest processing compared to I- and P-frames, as ME/DE is done using multiple reference frames/views. The energy consumption is also related to the content of the video, where a view or a set of frames may require less power due to easier-to-encode video content (*e.g.*, video content with lower motion intensity). In general, the work presented in Chapter 3 can be extended by including a power constraint in the formulated problem.

Then, in Chapter 4, we have investigated the problem of IMV in heterogeneous networks, proposing an adapting solution that allows users with different bandwidth capabilities to enjoy IMV at the maximum possible quality. In our simulations we only considered views independently encoded, not exploiting the interview correlation across the views. Therefore, another interesting extension of our problem is to consider inter-view coding in the layered multiview representation, where, for instance, inter-view coding is limited to views in the same or lower layers. This is certainly a challenging problem, as inter-view dependencies may bring exponential complexity in the DP algorithm proposed.

The development of a simple and effective view synthesis distortion model, able to quantify the impact of the texture and the depth maps of views used as references during the view synthesis, is crucial in order to further reduce the complexity of rate allocation algorithms, avoiding the multiple encodings of texture and depth maps. It is not an easy problem, as they depend on the particular scene geometry characteristics of the multiview videos. In this thesis, we have proposed a distortion model for the synthesized views that has been used in our simulations. However, the parameters related to the proportion of pixels disoccluded in

the projection of the reference views in the virtual view position are estimated from available texture and depth maps. Thus, it would be interesting to extend this work and model these parameters.

Finally, we have dealt with the rate allocation problem in Chapter 5. We have proposed a rate allocation algorithm where inter-view prediction is allowed. However, due to complexity reasons we have assumed that the effect of inter-view dependencies is limited to the first reference view of any predictively encoded view. Therefore, a possible extension of this work is to consider the effect of all previous reference views. Moreover, we also assumed that a view has only one reference view, which corresponds to the closest one. Thus, another interesting future research direction would be to allow more than one reference view (*e.g.*, B-frames) and optimize their selection. Such extensions would require efficient algorithms due to the high complexity of the problem.

The main emphasis of recent works on IMV has been on linear camera arrangements, where navigation is limited to horizontal displacements. Thus, a 3D navigation where camera and virtual views are available for an horizontal and vertical navigation, as well as allowing the user to zoom into areas of interest, is an exciting research topic. This requires new coding and view synthesis strategies as reference views for prediction or synthesis may not be horizontally aligned.

An additional limitation of current IMV systems is the narrow navigation range provided to the users. First, the number of camera views that can be transmitted is constrained by the network bandwidth. Second, the quality of synthesized images using IBR or DIBR techniques is determined by the distance between reference and virtual viewpoints. Thus, new rendering techniques are needed in order to improve the view synthesis quality and increase the navigation window offered to the user by using a small set of views. For instance, the plenoptic function [100] can be further exploited for image based-representations. The plenoptic function describes the intensity of each light ray in the world as a function of viewing angle, wavelength, time and viewing position. It captures everything that can potentially be seen by an optical device. Thus, by better exploiting the plenoptic function the number of view samples required to render a wider navigation domain can be better optimized to provide an improved user interaction. Recent solutions considering the plenoptic function are based on the use of multiple cameras, called super multiview video (SMV), and on the use of a single holoscopic camera that has an array of microlenses producing images with slightly different viewing angles. However, a novel imaging representation based on the plenoptic function will require the processing and transmission of a huge amount of information, meaning that efficient coding solutions are needed.

Overall, IMV will find many applications in different fields such as sports, advertising, education, design, exhibition, medicine, surveillance and so on. Thus, a rapid progress of the different stages of a processing chain of IMV, such as capturing, coding, transmission and display are needed to accelerate the introduction of this exciting technology.

A

Virtual View Distortion Model

In this appendix, we present an analytical model of the distortion of a rendered virtual view, where texture and depth maps quality information of the reference views are considered. This distortion model has been used for simulations in Chapter 4.

At the decoder side, if a requested view $u \in \mathscr{U}$ is not available, then it needs to be synthesized. We consider the *depth-image-based rendering* (DIBR) technique to render a view $u \in \mathscr{U}$, using the closest available right and left reference texture and associated depth maps, $v_R = \{v_R^t, v_R^d\}$ and $v_L = \{v_L^t, v_L^d\}$, for $(v_R, v_L) \in \mathscr{V}$. First, for each reference view, each pixel (x, y) is projected into the virtual view position (x', y'). These projected pixels, from the right and left reference views, form the textures $\hat{v}_{R,u}^t$ and $\hat{v}_{L,u}^t$, respectively. We follow a similar approach to the one in [101], where one of the reference views is considered as the dominant view. In particular, we first consider the pixels projected from the closest reference view to the virtual viewpoint. This view is denoted as v_1^t, for $v_1^t \in \{v_R^t, v_L^t\}$, and its projection as $\hat{v}_{1,u}^t$. Then, the missing pixels in $\hat{v}_{1,u}^t$ are filled from the projection of the second reference view, $\hat{v}_{2,u}^t$.

Note also that some pixels may not be available from any of the reference views, due to rounding error and/or disocclussions, these pixels are filled with inpainting methods [102]. In our model, a simple inpainting approach based on the interpolation of the neighboring available pixel values is assumed.

Overall, for each pixel (x,y) of the virtual view u, we have:

$$u(x,y) = \begin{cases} \hat{v}^t_{1,u}(x,y) & \text{if} \quad (x,y) \in (1-\alpha)u \\ \hat{v}^t_{2,u}(x,y) & \text{if} \quad (x,y) \in (1-\gamma)\alpha u \\ i(x,y) & \text{if} \quad (x,y) \in \gamma\alpha u \end{cases} \tag{A.1}$$

where $i(x,y)$ refers to the inpainting at pixel position (x,y), α denotes the proportion of pixels disoccluded in the closest reference view projection, and γ the proportion of pixels from αu that are not available in neither the right nor the left reference view projection.

This leads to the following virtual view distortion model:

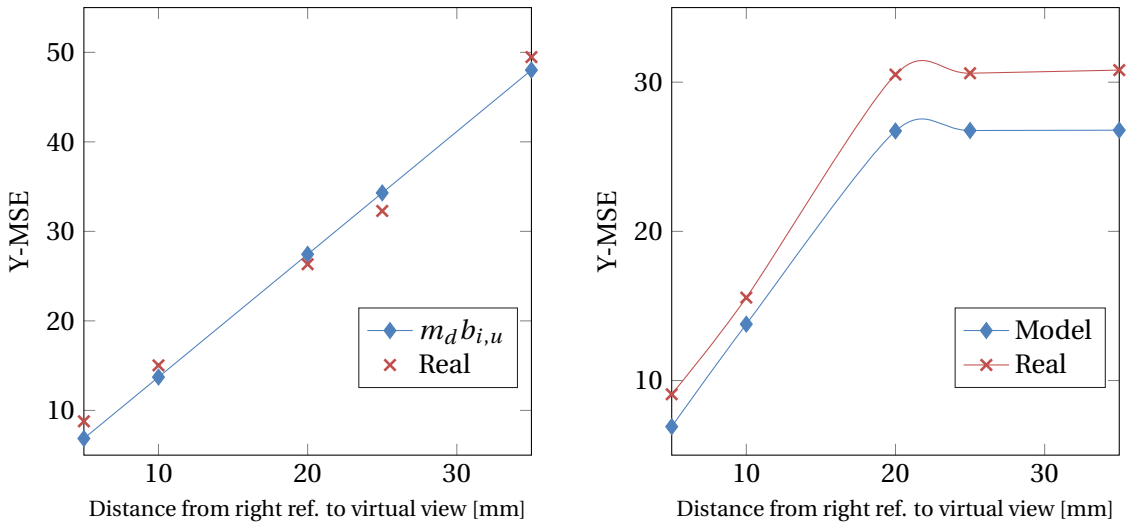
$$d_u(v_L,v_R) = (1-\alpha)\left(d_{\hat{v}^t_{1,u}}(v_L,v_R) + d_{\hat{v}^d_{1,u}}(v_L,v_R)\right) + (1-\gamma)\alpha\left(d_{\hat{v}^t_{2,u}}(v_L,v_R) + d_{\hat{v}^d_{2,u}}(v_L,v_R)\right) + \gamma\alpha\mathscr{I} \tag{A.2}$$

where, $d_{\hat{v}^t_{i,u}}$ and $d_{\hat{v}^d_{i,u}}$, for $i \in \{1,2\}$, denote the average distortion per pixel due to texture and to depth map errors, respectively. The average distortion per pixel in the inpainted areas is denoted by $\mathscr{I}$, which is assumed to take a constant value that only depends on the scene content. The proportion of disoccluded pixels, α and γ, are obtained from the depth maps of the reference views, which are available at the sender side.

As pixel intensity values are copied from the reference views to their projections, the distortion of the projected views, $d_{\hat{v}^t_{i,u}}$, corresponds to the distortion of the reference views $d_{v^t_i}$, which can be modeled in terms of the rate as $\sigma^2 2^{-2R}$ [103]. Then, we can assume that $d_{\hat{v}^t_{1,u}} = d_{v^t_1}$ and $d_{\hat{v}^t_{2,u}} = d_{v^t_2}$, simplifying the notation of (A.2).

Depth maps errors accounts for position errors, and it has been shown that the distortion value of the projected image linearly increases with the distance to the virtual view u [104]. Therefore, in this work, $d_{\hat{v}^d_{1,u}}$ and $d_{\hat{v}^d_{2,u}}$, are linearly modeled as a function of the distance to the reference view, i.e., $d_{\hat{v}^d_{1,u}} = m_d \cdot b_{1,u}$, where, $b_{1,u}$ stands for the baseline distance between virtual view u and reference view v_1, while m_d is the growing rate of the distortion of the projected view. The distortion $d_{\hat{v}^d_{2,u}}$ is similarly defined. In this work, we opt for depth maps encoded at low compression ratio (high quality), since they contribute with a small proportion of the overall rate, compared to texture data. Thus, we only consider the distortion due to errors originally present in the depth maps, due to capturing or estimation error. In order to simplify the notation of (A.2), we write $d_{\hat{v}^t_{i,u}}$, $d_{\hat{v}^d_{i,u}}$ as $d_{v^t_i}$, $d_{v^d_i}$, when refererring to the distortion model.

In Fig. A.1 the distortion model is illustrated using the image dataset *Bikes* [4], where views



(a) (b)

Figure A.1 – Distortion model illustration for *Bikes* [4] image dataset, for a virtual view $u = 43$, right reference views $v_R = \{44, 45, 47, 48, 50\}$ and fixed left reference view $v_L = 40$. (a) Comparison of modeled and real distortion of the right view projection due to depth map $d_{v_i^d} = m_d b_{i,u}$, with $m_d = 1.372$. (b) Comparison of virtual view distortion modeled and real by fixing the left reference view $v_L = 40$.

have a baseline separation of $5mm$. We consider view $u = 43$ as a virtual view, synthesized using a set of possible right reference views $v_R = \{44, 45, 47, 48, 50\}$, while the left reference view is kept fixed $v_L = 40$. First, the linear behavior of the distortion of the right view projection due to depth map $d_{v_i^d}$ is shown in Fig. A.1a, where $m_d = 1.372$. Both the modeled and real values of the distotion of virtual view $u = 43$ is presented in Fig. A.1b when different right reference views are used. Although, the distortion values obtained by the proposed model are not close enough to the real values, the model is able to capture the behaviour of the distortion, which has proven to be sufficient for rate allocation problems as the one tackled in Chapter 4. Further studies will be necessary to have a model that better fits the distortion of a synthesized view.

B

Proof of Optimality

In this appendix, we prove that by solving the equivalent Lagrangian unconstrained problem (5.7) of the original constrained formulation (5.3) we are able to find the optimal solution.

Lemma 1: if an optimal solution $(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ to the unconstrained Lagrangian problem corresponding to multiplier value λ^* satisfies the rate constraint exactly, *i.e.*,

$$R(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) = B, \tag{B.1}$$

then, $(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ is also the optimal solution to the original constrained problem.

Proof: The optimality of the solution $(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ implies:

$$D(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) + \lambda^* R(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) \leq D(\mathcal{V}, \mathbf{q}) + \lambda^* R(\mathcal{V}, \mathbf{q}), \quad \forall \{\mathcal{V}, \mathbf{q}\} \tag{B.2}$$

Rearranging the terms, we get:

$$\begin{aligned}\lambda^* \left[R(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) - R(\mathcal{V}, \mathbf{q})\right] &\leq D(\mathcal{V}, \mathbf{q}) - D(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) \\ \lambda^* \left[B - R(\mathcal{V}, \mathbf{q})\right] &\leq D(\mathcal{V}, \mathbf{q}) - D(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) \end{aligned} \tag{B.3}$$

Now we restrict our solution space to a subspace $\mathcal{S}$ where $R(\mathcal{V}, \mathbf{q}) \leq B$. Then,

$$\begin{aligned} 0 \leq \lambda^* \left[B - R(\mathcal{V}, \mathbf{q})\right] &\leq D(\mathcal{V}, \mathbf{q}) - D(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}), \;\; \forall (\mathcal{V}, \mathbf{q}) \in \mathcal{S} \\ D(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*}) &\leq D(\mathcal{V}, \mathbf{q}), \;\; \forall (\mathcal{V}, \mathbf{q}) \in \mathcal{S} \end{aligned} \tag{B.4}$$

We can thus conclude that $(\mathcal{V}_{\lambda^*}, \mathbf{q}_{\lambda^*})$ is an optimal solution to the original constrained problem as well. □

C

Performance Bound

We now prove the performance bound given in Eq. (5.10) of Chapter 5.

Let $(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ and $(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2})$ be two solutions of the problem in (5.7) using λ_1 and λ_2 with resulting rates:

$$R(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B < R(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \tag{C.1}$$

We can derive a performance bound for feasible solution $(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$ as follows. Denote by $(\mathcal{V}^*, \mathbf{q}^*)$ the optimal solution to the original constrained problem. By the optimality of $(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2})$, we can write:

$$\begin{aligned} 0 \leq \lambda^* \left[R(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) - R(\mathcal{V}^*, \mathbf{q}^*) \right] &\leq D(\mathcal{V}^*, \mathbf{q}^*) - D(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \\ D(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) &\leq D(\mathcal{V}^*, \mathbf{q}^*) \end{aligned} \tag{C.2}$$

where the second line is true because $B < R(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2})$ and $R(\mathcal{V}^*, \mathbf{q}^*) \leq B$. By the optimality of

$(\mathcal{V}^*, \mathbf{q}^*)$, we also know that:

$$D(\mathcal{V}^*, \mathbf{q}^*) \leq D(\mathcal{V}, \mathbf{q}), \quad \forall (\mathcal{V}, \mathbf{q}) \in \mathcal{S} \tag{C.3}$$

where, $\mathcal{S}$ denotes the set of solutions that have a total rate lower than B. Note that $\mathcal{S}$ includes $(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1})$, since $R(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) < B$. Combining the inequalities in (C.2) and (C.3), we can write:

$$\begin{aligned} D(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \leq D(\mathcal{V}^*, \mathbf{q}^*) \leq D(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) \\ \left| D(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathcal{V}^*, \mathbf{q}^*) \right| \leq \left| D(\mathcal{V}_{\lambda_1}, \mathbf{q}_{\lambda_1}) - D(\mathcal{V}_{\lambda_2}, \mathbf{q}_{\lambda_2}) \right| \end{aligned} \tag{C.4}$$

which concludes the proof. □

Bibliography

[1] Shark multiview sequence. [Online]. Available: http://www.fujii.nuee.nagoya-u.ac.jp/NICT/NICT.htm

[2] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 600–608, August 2004.

[3] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, K. Wegner, and M. Wildeboer, "Poznań multiview video test sequences and camera parameters," in *ISO/IEC JTC1/SC29/WG11 MPEG 2009/M17050*, Xian, China, October 2009.

[4] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Trans. on Graphics*, vol. 32, no. 4, pp. 73:1–73:12, July 2013.

[5] M. Tanimoto, "FTV standardization for super-multiview and free navigation in MPEG," in *Proc. of SPIE*, vol. 9495, 2015, pp. 94 950T–94 950T–13.

[6] M. Tanimoto, T. Senoh, S. Shimizu, S. Naito, M. Domanski, A. Vetro, and M. Preda, "Proposal on a new activity for the third phase of FTV," in *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, Vienna, Austria, July 2013.

[7] M. Tanimoto, "Overview of FTV (free-viewpoint television)," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, Piscataway, NJ, USA, June 2009, pp. 1552–1553.

[8] ——, "FTV (free-viewpoint television) for ray and sound reproducing in 3D space," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, March 2012, pp. 5441–5444.

[9] B. Li, H. Li, L. Li, and J. Zhang, "Rate control by R-lambda model for HEVC, JCTVC-K0103," Joint Collaborative Team on Video Coding (JCT-VC), Changai, China, October 2012.

[10] W. Lim, H. Jo, and D. Sim, "Inter-view MV-based rate prediction for rate control of 3D multi-view video coding, JCT3V-F0166," Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, October 2013.

[11] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview imaging and 3DTV," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 10–21, November 2007.

[12] M. P. Tehrani, S. Shimizu, G. Lafruit, T. Senoh, T. Fujii, A. Vetro, and M. Tanimoto, "Use cases and requirements on free-viewpoint television (FTV)," in *ISO/IEC JTC1/SC29/WG11 MPEG2013/N14104*, Geneva, Switzerland, October 2013.

[13] P. Na Bangchang, T. Fujii, and M. Tanimoto, "Experimental system of free viewpoint television," in *Proc. of SPIE*, vol. 5006, 2003, pp. 554–563.

[14] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound - 100-camera and microphone system," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, July 2006, pp. 437–440.

[15] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*, 1st ed. Wiley Publishing, 2013.

[16] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (tof) cameras: A survey," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, September 2011.

[17] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Journal of Signal Processing: Image Communications*, vol. 22, no. 2, pp. 217–234, February 2007.

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

[19] J.-G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in *Proc. of ACM Int. Conf. on Multimedia*, ser. MULTIMEDIA '05. New York, NY, USA: ACM, 2005, pp. 161–170.

[20] J. Chakareski, "Wireless streaming of interactive multi-view video via network compression and path diversity," *Communications, IEEE Transactions on*, vol. 62, no. 4, pp. 1350–1357, April 2014.

[21] D. Ren, S.-H. Chan, G. Cheung, and P. Frossard, "Coding structure and replication optimization for interactive multiview video streaming," *IEEE Trans. on Multimedia*, vol. 16, no. 7, pp. 1874–1887, November 2014.

[22] T. Kanade, P. J. Narayanan, and P. W. Rander, "Virtualized reality: Concepts and early results," in *Proceedings of the IEEE Workshop on Representation of Visual Scenes*, ser. VSR '95, Washington, DC, USA, 1995, pp. 69–.

[23] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer, "A survey of methods for volumetric scene reconstruction from photographs," in *Proceedings of the 2001 Eurographics Conference on Volume Graphics*, ser. VG'01, Aire-la-Ville, Switzerland, 2001, pp. 81–101.

[24] S. Chan, H.-Y. Shum, and K.-T. Ng, "Image-based rendering and synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 22–33, Nov 2007.

[25] S. B. Kang and H.-Y. Shum, "A review of image-based rendering techniques." Institute of Electrical and Electronics Engineers, Inc., June 2000.

[26] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, November 2003.

[27] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 31–42.

[28] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '93, New York, NY, USA, 1993, pp. 279–288.

[29] W. R. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Proc. of the Symposium on Interactive 3D Graphics*, ser. I3D '97, New York, NY, USA, 1997, pp. 7–ff.

[30] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *Proc. of SPIE*, vol. 7443, 2009, pp. 74 430T–74 430T–11.

[31] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Journal of Signal Processing: Image Communications*, vol. 24, no. 1–2, pp. 73–88, 2009, special issue on advances in three-dimensional television and video.

[32] K. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Trans. on Image Processing*, vol. 16, no. 2, pp. 545–553, February 2007.

[33] E. Kurutepe, M. Civanlar, and A. Tekalp, "Client-driven selective streaming of multiview video for interactive 3DTV," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1558–1565, November 2007.

[34] C. Zhang and D. Florencio, "Joint tracking and multiview video compression," in *Proc. of SPIE on Visual Communications and Image Processing*, July 2010.

[35] G. Cheung, A. Ortega, and T. Sakamoto, "Coding structure optimization for interactive multiview streaming in virtual world observation," in *Proc. of IEEE MMSP*, Cairns, Queensland, Australia, October 2008.

[36] X. Xiu, G. Cheung, and J. Liang, "Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis," *IEEE Trans. on Multimedia*, vol. 14, no. 4, pp. 1109–1126, August 2012.

[37] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV: A survey," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1606–1621, November 2007.

[38] G. Akar, A. Tekalp, C. Fehn, and M. Civanlar, "Transport methods in 3dtv: A survey," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1622–1630, Nov 2007.

[39] ITU-T and I. J. 1, "Advanced video coding for generic audiovisual services," *ITU-T Recommendation H.264*, February 2014.

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

[41] A. Kondoz and T. Dagiuklas, *3D Future Internet Media*. Springer Publishing Company, Incorporated, 2013.

[42] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. of the IEEE*, vol. 99, no. 4, pp. 626–642, April 2011.

[43] G. Sullivan, J. Boyce, Y. Chen, J.-R. Ohm, C. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, December 2013.

[44] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461–1473, November 2007.

[45] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *Proc. of IEEE Int. Conf. on Image Processing*, San Antonio, TX, October 2007.

[46] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. on Image Processing*, vol. 22, no. 9, pp. 3366–3378, September 2013.

[47] L. Do, S. Zinger, Y. Morvan, and P. de With, "Quality improving techniques in DIBR for free-viewpoint video," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, May 2009, pp. 1–4.

[48] J.-H. Kim, J. Garcia, and A. Ortega, "Dependent bit allocation in multiview video coding," in *Proc. of IEEE Int. Conf. on Image Processing*, Genoa, Italy, September 2005.

[49] G. Cheung, V. Velisavljevic, and A. Ortega, "On dependent bit allocation for multiview image coding with depth-image-based rendering," *IEEE Trans. on Image Processing*, vol. 20, no. 11, pp. 3179–3194, November 2011.

[50] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," in *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 36, no.9, September 1988, pp. 1445–1453.

[51] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," in *IEEE Trans. on Image Processing*, vol. 3, no.5, September 1994.

[52] S. Liu and C.-C. J. Kuo, "Joint temporal-spatial bit allocation for video coding with dependency," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no.1, January 2005, pp. 15–26.

[53] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, November 1998.

[54] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[55] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[56] M. Drose, C. Clemens, and T. Sikora, "Extending single-view scalable video coding to multi-view based on H.264/AVC," in *Proc. of IEEE Int. Conf. on Image Processing*, October 2006, pp. 2977–2980.

[57] N. Ozbek and A. Tekalp, "Scalable multi-view video coding for interactive 3DTV," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, July 2006, pp. 213–216.

[58] J. Chakareski, V. Velisavljevic, and V. Stankovic, "User-action-driven view and rate scalable multiview video coding," *IEEE Trans. on Image Processing*, vol. 22, no. 9, pp. 3473–3484, September 2013.

[59] V. Velisavljevic, V. Stankovic, J. Chakareski, and G. Cheung, "View and rate scalable multiview image coding with depth-image-based rendering," in *Proc. of IEEE DSP*, July 2011, pp. 1–8.

[60] J. Chakareski, V. Velisavljevic, and V. Stankovic, "View-popularity-driven joint source and channel coding of view and rate scalable multi-view video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 474–486, April 2015.

[61] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "RD-optimized interactive streaming of multiview video with multiple encodings," *Journal Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 523–532, July 2010.

[62] H. Kimata, M. Kitahara, K. Kamikura, Y. Yashima, T. Fujii, and M. Tanimoto, "System design of free viewpoint video communication," in *Int. Conf. on Computer and Information Technology*, Wuhan, China, September 2004.

[63] T. Fujihashi, Z. Pan, and T. Watanabe, "UMSM: A traffic reduction method on multiview video streaming for multiple users," *IEEE Trans. on Multimedia*, vol. 16, no. 1, pp. 228–241, January 2014.

[64] T. Maugey, I. Daribo, G. Cheung, and P. Frossard, "Navigation domain representation for interactive multiview imaging," *IEEE Trans. on Image Processing*, vol. 22, no. 9, pp. 3459–3472, September 2013.

[65] Y. Liu, Q. Dai, Z. You, and W. Xu, "Rate-prediction structure complexity analysis for multi-view video coding using hybrid genetic algorithms," in *Proc. of SPIE*, San Jose, CA, USA, January 2007.

[66] F. Chen, D. Delannay, and C. De Vleeschouwer, "An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study," *IEEE Trans. on Multimedia*, vol. 13, no. 6, pp. 1381–1394, December 2011.

[67] A. Fiandrotti, J. Chakareski, and P. Frossard, "Popularity-aware rate allocation in multiview video coding," in *Proc. of IEEE VCIP*, Huang Shan, An Hui, China, July 2010, invited paper.

[68] K. Klimaszewski, K. Wegner, and M. Domanski, "Distortions of synthesized views caused by compression of views and depth maps," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Potsdam, Germany, May 2009.

[69] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, Toronto, Ontario, Canada, July 2006.

[70] K. Muller, A. Smolic, K. Dix, P. Merkle, and T. Wiegand, "Coding and intermediate view synthesis of multiview video plus depth," in *Proc. of IEEE Int. Conf. on Image Processing*, Cairo, Egypt, November 2009.

[71] E. Bosc, V. Jantet, M. Pressigout, L. Morin, and C. Guillemot, "Bit-rate allocation for multiview video plus depth," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Antalya, Turkey, May 2011.

[72] E. Bosc, P. Riou, M. Pressigout, and L. Morin, "Bit-rate allocation between texture and depth: Influence of data sequence characteristics," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Zurich, Switzerland, October 2012.

[73] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.

[74] JMVC 8.2 software. [Online]. Available: garcon.ient.rwth-aachen.de

[75] HTM 6.2 software. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-6.2/

[76] Tanimoto laboratory test sequences for mvc-ftv. [Online]. Available: http://www.tanimoto.nuee.nagoya-u.ac.jp/

[77] I. Feldmann, M. Mueller, F. Zilly, R. Tanger, K. Mueller, A. Smolic, P. Kauff, and T. Wiegand, "HHI test material for 3d video," in *ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15413*, Archamps, France, April 2008.

[78] J. Zhang, R. Li, H. Li, D. Rusanovskyy, and M. Hannuksela, "Ghost town fly 3DV sequence for purposes of 3DV standardization," in *ISO/IEC JTC1/SC29/WG11 MPEG 2010/M20027*, Geneva, Switzerland, March 2011.

[79] D. Rusanovskyy, P. Aflaki, and M. M. Hannuksela, "Undo dancer 3DV sequence for purposes of 3DV standardization," in *ISO/IEC JTC1/SC29/WG11 MPEG 2010/M20028*, Geneva, Switzerland, March 2011.

[80] D. Rusanovskyy, K. Müller, and A. Vetro, "Common test conditions of 3DV core experiments," in *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, Geneva,Switzerland, October 2013.

[81] M. Tanimoto, T. Fujii, and K. Suzuki, "Depth estimation reference software (DERS) 5.0," in *ISO/IEC JTC1/SC29/WG11 M16923*, Xian, China, October 2009.

[82] ——, "View synthesis algorithm in view synthesis reference software 2.0 (VSRS 2.0)," in *ISO/IEC JTC1/SC29/WG11 M16090*, Lausanne, Switzerland, February 2008.

[83] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Proc. of SPIE*, vol. 5291, May 2004, pp. 93–104.

[84] L. Toni, N. Thomos, and P. Frossard, "Interactive free viewpoint video streaming using prioritized network coding," in *Proc. of IEEE MMSP*, Pula, Italy, September 2013.

[85] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, no. 12, December 2012.

[86] P. S. P. Wang, *Pattern Recognition, Machine Intelligence and Biometrics*. Springer Publishing Company, Incorporated, 2011.

[87] R. Lagendijk, E. D. Frimout, and J. Biemond, "Low-complexity rate-distortion optimal transcoding of MPEG I-frames," *Journal of Signal Processing: Image Communications*, vol. 15, pp. 531–544, March 2000.

[88] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.

[89] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Trans. on Image Processing*, vol. 9, no. 3, pp. 340–356, March 2000.

[90] Y.-M. Chen, I. Bajic, and P. Saeedi, "Coarse-to-fine moving region segmentation in compressed video," in *Workshop on Image Analysis for Multimedia Interactive Services*, May 2009, pp. 45–48.

[91] Y.-M. Chen and I. Bajic, "Compressed-domain moving region segmentation with pixel precision using motion integration," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, August 2009, pp. 442–447.

[92] P. Goorts, S. Maesen, M. Dumont, S. Rogmans, and P. Bekaert, "Free viewpoint video for soccer using histogram-based validity maps in plane sweeping," in *Proc. of Int. Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2014, pp. 378–386.

[93] K. Müller and A. Vetro, "Common test conditions of 3DV core experiments, JCT3V-G1100," Joint Collaborative Team on Video Coding (JCT-VC), San José, US, January 2014.

[94] L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7431, pp. 447–457.

[95] J. Zhai, K. Yu, J. Li, and S. Li, "A low complexity motion compensated frame interpolation method," in *Proc. of IEEE Int. Symposium on Circuits and Systems*, Kobe, Japan, May 2005.

[96] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Istanbul, Turkey, May 2008.

[97] HM 15.0 software. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-15.0-dev/

[98] HTM 13.0 software. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-13.0/

[99] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. on Image Processing*, vol. 22, no. 9, pp. 3366–3378, September 2013.

[100] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. MIT Press, 1991, pp. 3–20.

[101] Y. Zhao, C. Zhu, Z. Chen, D. Tian, and L. Yu, "Boundary artifact reduction in view synthesis of 3D video: From perspective of texture-depth alignment," *IEEE Trans. on Broadcasting*, vol. 57, no. 2, pp. 510–522, June 2011.

[102] Z. Tauber, Z.-N. Li, and M. S. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Trans.on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 4, pp. 527–540, July 2007.

[103] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 1991.

[104] K. Muller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *The Institute of Electrical and Electronics Engineers*, vol. 99, no. 4, pp. 643–656, April 2011.

Ana **De Abreu**

Avenue Druey 14B
CH-1018 Lausanne
+41 (0) 76 454 71 89
ana.deabreug@gmail.com

Education

07.2011 - 10.2015 **PhD, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland**.
- Thesis: Optimizing Coding Strategies for Interactive Multi-view Video.
- Work supervison: Prof. Pascal Frossard and Prof. Fernando Pereira
- Double degree program with **Instituto Superior Técnico** (IST), Lisboa.

09.2007 - 11.2009 **MSc, Telecommunications Engineering, Politecnico di Torino, Italy**.
- Thesis: Scalable Video Coding for Multimedia Streaming Over the Internet.
- Work supervison: Prof. C. -C. J. Kuo and Prof. Enrico Magli
- GPA: 110/110

09.2003 - 11.2009 **BSc, Electrical Engineering**, *Central University of Venezuela, Venezuela*.
- Double degree program with Politecnico di Torino, Italy.
- Engineering Faculty Honor List. Graduated first in the 2009 class.

Experience

07.2011 - 10.2015 **École Polytechnique Fédérale de Lausanne (EPFL)**, *Switzerland*.
Research assistant.
- Proposed and designed multiview video solutions for bandwidth constrained networks gaining vast knowledge in Algorithms, Optimization, Networking, Video Quality, Video Coding and Systems.
- Involved in QoSTREAM project in 2013, 2014.
- Involved in international collaborations (Japan, England, France, Portugal).
- Participated in international conferences and journals (complete list).
- Worked as a teaching assistant of Image Communication course.
- Led students and proposed projects in image quality model for view synthesis, multiview video transmission in Dash, and video coding.

05.2010 - 02.2011 **Cisco System**, Switzerland.
Intern for video quality assessment.
- Developed and executed tests for video quality assessment products that monitor videos accross dynamic IP networks.
- Designed and implemented Unix/Linux based test bed using Cisco 2800 Series Routers.
- Documented test plans, reported defects, and tracked test execution status. Wrote a technical report at the end of the project.

05.2010 - 04.2011 **École Polytechnique Fédérale de Lausanne (EPFL)**, Switzerland.
Research Assistant.
- Proposed and designed a full algorithm based on Network Coding for the passive network topology inference problem.

03.2009 - 09.2009 **University of Southern California (USC)**, Los Angeles, USA.
Intern for Master thesis project.

- Proposed an unequal error protection scheme for scalable video transmission over error prone channels to increase the end-to-end video quality.
- Awarded with a research fellowship by Politecnico di Torino for thesis abroad.

Technical skills

Video	HEVC, H.264, 3D-HEVC, Multiview Video Coding (MVC), depth estimation reference software (DERS), view synthesis reference software (VSRS).
Programming	Matlab, C/C++, Java, Linux and UNIX shells and tools.
Networking	Network sniffer (tcpdump, ethereal), network simulator NS3.

Languages

- Spanish – Mother tongue.
- English – Fluent (C1).
- French – Intermediate Level (B1).
- Italian – Intermediate Level (B1).
- Portuguese – Intermediate Level (B1).

Publications

Journal Papers.

- A.De Abreu, P. Frossard and F. Pereira; *"Optimized MVC Prediction Structures for Interactive Multiview Video Streaming"*, IEEE Signal Processing Letters, vol. 20, no. 6, pp. 603-606, June 2013.
- A.De Abreu, P. Frossard and F. Pereira; *"Optimized Multiview video Plus Depth Prediction Structures for Interactive Multiview Video Streaming"*, IEEE Journal of Selected Topics in Signal Processing, vol 9, no. 3, pp. 487-500, April 2015.
- A.De Abreu, L. Toni, N. Thomos, T. Maugey, F. Pereira and P. Frossard; *"Optimal Layered Representation for Adaptive Interactive Multiview Video Streaming"*, Journal of Visual Communication and Image Representation, accepted under minor revision.

Conference Papers.

- A. De Abreu, L. Tony, T. Thomas, N. Thomos, P. Frossard, F. Pereira; *"Multiview Video Representations for Quality-Scalable Navigation"*, IEEE Conference on Visual Communications and Image Processing (VCIP); Valeta, Malta; 7-10 Dec. 2014 .
- A.De Abreu, P. Frossard and F. Pereira; *"Optimized Multiview video Plus Depth Prediction Structures for Interactive Multiview Video Streaming"*, Picture Coding Symposium (PCS); San Jose, CA, US; 8-13 December 2013.

Other activities

10.2006 - 07.2007 **Volunteer Experience**, Venezuela.
Teacher support in limited resources schools

- Assisting teachers covering partial tuition hour.
- Leader of volunteer group for children under the age of 10.

09.2012 - 07.2013 **Toastmasters UNIL-EPFL club**, Switzerland.
Participating member

- Organized meetings, took different roles in the various meetings including oral presentations.