TOWARDS VISUALIZATION AND SEARCHING: A DUAL-PURPOSE VIDEO CODING APPROACH

Renam Castro da Silva

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Rio de Janeiro
Fevereiro de 2018

TOWARDS VISUALIZATION AND SEARCHING: A DUAL-PURPOSE VIDEO CODING APPROACH

Renam Castro da Silva

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D

Prof. Fernando Manuel Bernardo Pereira, Ph.D

Prof. Sergio Lima Netto, Ph.D

Prof. Ricardo Lopes de Queiroz, Ph.D

Prof. Touradj Ebrahimi, Ph.D

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2018

Silva, Renam Castro da

Towards Visualization and Searching: a Dual-Purpose Video Coding Approach/Renam Castro da Silva. – Rio de Janeiro: UFRJ/COPPE, 2018.

XVII, 134 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2018.

Referências Bibliográficas: p. 124 – 134.

1. Visualization. 2. Searching. 3. Dual-purpose video coding. 4. HEVC. 5. Keypoint. 6. Descriptor. 7. Matching. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

For the readers

Acknowledgments

I am very thankful to my supervisors prof. Eduardo Silva and prof. Fernando Pereira for their guidance, support, patience, and encouragement. It has been an edifying experience to work with such motivated, engaged and outstanding researchers. Thank you for the technical discussions; for providing ideas, suggestions, and feedbacks that made possible the development of this research; for your dedication in reviewing my work; for leading by example. I will be always in debt to you.

Also, I would like to thank the professors, colleagues and staff from the Signal, Multimedia and Telecommunications Laboratory (SMT) of Universidade Federal do Rio de Janeiro for providing me the technical education, friendship and the facilities for carrying out my graduate studies. I also would like to express gratitude to the professors and colleagues from the Multimedia Signal Processing (MSP) Group from Instituto de Telecomunicações (IT) for receiving me as visiting student.

I also thank the Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for their financial support.

I would like to express my deepest gratitude to my family for the support, care and love. Especially, my grandmother Francisca who despite the simple life, seems to have accumulated a lot of life experience and wisdom.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

UMA ABORDAGEM DE CODIFICAÇÃO DE VÍDEO DE DUPLA FINALIDADE: VISUALIZAÇÃO E BUSCA VISUAL

Renam Castro da Silva

Fevereiro/2018

Orientadores: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Programa: Engenharia Elétrica

Em modernas aplicações de vídeo, o papel do vídeo decodificado é muito mais que simplesmente preencher uma tela para visualização. Para oferecer aplicações mais poderosas por meio de sinais de vídeo, é cada vez mais crítico não apenas considerar a qualidade do conteúdo objetivando sua visualização, mas também possibilitar meios de realizar busca por conteúdos semelhantes. Requisitos de visualização e de busca são considerados, por exemplo, em modernas aplicações de vídeo vigilância e comunicações pessoais. No entanto, as atuais soluções de codificação de vídeo são fortemente voltadas aos requisitos de visualização. Nesse contexto, o objetivo deste trabalho é propor uma solução de codificação de vídeo de propósito duplo, objetivando tanto requisitos de visualização quanto de busca. Para isso, é proposto um arcabouço de codificação em que a abordagem usual de codificação de píxeis é combinada com uma nova abordagem de codificação baseada em *features* visuais. Nessa solução, alguns quadros são codificados usando um conjunto de pares de *keypoints* casados, possibilitando não apenas visualização, mas também provendo ao decodificador valiosas informações de *features* visuais, extraídas no codificador a partir do conteúdo original, que são instrumentais em aplicações de busca. A solução proposta emprega um esquema flexível de otimização Lagrangiana onde o processamento baseado em píxel é combinado com o processamento baseado em *features* visuais objetivando encontrar um compromisso adequado entre os desempenhos de visualização e de busca. Os resultados experimentais mostram a flexibilidade da solução proposta em alcançar diferentes compromissos de otimização, nomeadamente desempenho competitivo em relação ao padrão HEVC tanto em termos de visualização quanto de busca.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

TOWARDS VISUALIZATION AND SEARCHING: A DUAL-PURPOSE VIDEO CODING APPROACH

Renam Castro da Silva

February/2018

Advisors: Eduardo Antônio Barros da Silva
Fernando Manuel Bernardo Pereira

Department: Electrical Engineering

In modern video applications, the role of the decoded video is much more than filling a screen for visualization. To offer powerful video-enabled applications, it is increasingly critical not only to visualize the decoded video but also to provide efficient searching capabilities for similar content. Video surveillance and personal communication applications are critical examples of these dual visualization and searching requirements. However, current video coding solutions are strongly biased towards the visualization needs. In this context, the goal of this work is to propose a dual-purpose video coding solution targeting both visualization and searching needs by adopting a hybrid coding framework where the usual pixel-based coding approach is combined with a novel feature-based coding approach. In this novel dual-purpose video coding solution, some frames are coded using a set of keypoint matches, which not only allow decoding for visualization, but also provide the decoder valuable feature-related information, extracted at the encoder from the original frames, instrumental for efficient searching. The proposed solution is based on a flexible joint Lagrangian optimization framework where pixel-based and feature-based processing are combined to find the most appropriate trade-off between the visualization and searching performances. Extensive experimental results for the assessment of the proposed dual-purpose video coding solution under meaningful test conditions are presented. The results show the flexibility of the proposed coding solution to achieve different optimization trade-offs, notably competitive performance regarding the state-of-the-art HEVC standard both in terms of visualization and searching performance.

Contents

List of Figures

List of Tables

List of Abbreviations

AC	Arithmetic Coding, p. 16
AMP	Asymmetric Motion Partitions, p. 22
AMVP	Advanced Motion Vector Prediction, p. 23
ATC	Analyze-then-Compress, p. 3
BD-PSNR	Bjontegaard Delta-PSNR, p. 29
BD-Rate	Bjontegaard Delta-Rate, p. 29
BRIEF	Binary Robust Independent Elementary Features, p. 46
BRISK	Binary Robust Invariant Scalable Keypoints, p. 37, 46
CABAC	Context-Adaptive Binary Arithmetic Coding, p. 16
CB	Coding Block, p. 18
CCD	Charge-Coupled Device, p. 9
CDVS	Compact Descriptors for Visual Search, p. 3
CHoG	Compressed Histogram of Gradient Orientation, p. 45
CTA	Compress-then-Analyze, p. 3
CTB	Coding Transform Block, p. 19
CU	Coding Unit, p. 19
DCT	Discrete Cosine Transform, p. 14
DPB	Decoded Picture Buffer, p. 21
DPVC	Dual-Purpose Video Coding, p. 62
DST	Discrete Sine Transform, p. 15

DoG	Difference of Gaussians, p. 38
EMD	Earth Mover's Distance, p. 49
FAST	Features From Accelerated Segment Test, p. 37
FREAK	Fast Retina Keypoint, p. 46
GLOH	Gradient Location and Orientation Histogram, p. 45
HATC	Hybrid-Analyze-then-Compress, p. 3
HD	High Definition, p. 85
HEVC	High Efficiency Video Coding, p. 2
HM	HEVC Test Model, p. 30
HVS	Human Visual System, p. 9
KL	Karhunen-Loève, p. 14
LoG	Laplacian of Gaussian, p. 38
MPEG	Moving Picture Expert Group, p. 2
MPM	Most Probable Modes, p. 21
PB	Prediction Block, p. 19
PDE	Partial Differential Equation, p. 108
PSNR	Peak Signal-to-Noise Ratio, p. 28
PU	Prediction Unit, p. 19
QP	Quantization parameter, p. 25
RD	Rate-Distortion, p. 29
RGB	Red, Green and Blue (color components), p. 9
RQT	Residual quadtree, p. 24
SAD	Sum of Absolute Differences, p. 28
SAO	Sample Adaptive Offset, p. 27
SIFT	Scale Invariant Feature Transform, p. 2

SSD	Sum of Squared Differences, p. 28
SSIM	Structure Similarity, p. 28
SURF	Speeded-Up Robust Features, p. 2
TB	Transform Block, p. 19
TU	Transform Unit, p. 19
VCEG	Video Coding Experts Group, p. 2
VSN	Visual Sensor Networks, p. 4

Chapter 1

Introduction

This chapter presents the motivation and context within which this research work has been developed, its objectives and main contributions. Also, it outlines the topics addressed in the upcoming chapters.

1.1 Context and motivation

Since the prehistoric period, humans realized the importance of visual information, in particular, its power and effectiveness to communicate. Take for instance the cave paintings made by the early humans. Tens of thousands of years later that rudimentary effort, a sophisticated set of technology has emerged to acquire, process, store, manipulate, share and deliver visual information.

Nowadays visual information is popularly consumed in the form of digital images and videos. Its applications range from medicine to entertainment, education, video surveillance, industrial inspection systems, scientific research, TV broadcasting and Internet-based streaming services. Such pervasive use creates heterogeneous requirements and constraints in the processing pipeline designed to handle visual data. Since resources are scarce, the need for efficient representation at a desired quality is commonly required for storage and transmission over bandwidth restricted networks.

The problem of minimizing the amount of bits to meet a target reconstruction quality is addressed in the image and video coding research. From a general point of view, image and video signals can be coded in a lossless or lossy manner. In lossless coding the compressed signal fully preserves the data of the original signal at the penalty of achieving only modest compression factors [1, 2]. Higher compression factors can be achieved by carefully allowing some loss of fidelity in the reconstructed signal. For instance, in transform coding, the energy compaction property facilitates discarding negligible information for the human visual system by a suitable quantization strategy, usually allowing higher distortion in the higher

frequency range. This work focus on lossy video coding. Image and video coding becomes even more challenging when considering other application requirements, *e.g.*, low computational cost, rate and quality scalability.

For decades the research community has been working on the design of efficient coding algorithms to handle the increasing amount of data in the form of image and video signals. Successive generations of video coding standards have been developed with increasing compression efficiency. The state-of-the-art in video coding technology is the High Efficiency Video Coding (HEVC) standard [3, 4], a product of the joint effort between ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Expert Group (MPEG). It adopts the block-based prediction and transform coding framework successfully employed in previous standards. HEVC was designed aiming to provide 50% of bit rate savings compared with the previous standard H.264/AVC [5] for the same perceptual quality. This is achieved with a set of rather flexible coding tools which are able to adapt to the content characteristics in order to obtain a very compact representation of the input signal. It offers highly efficient video coding solutions for a wide variety of applications, from video surveillance and personal communications to UHD television and streaming. HEVC and all the previous video coding standards adopt a pure pixel-based video coding approach, which essentially targets visualization capabilities and thus visual quality. However, with the increasing amount and omnipresence of digital video, users are increasingly not just visualizing the decoded video but also using it for other purposes, notably searching for similar visual content. This is happening in many application domains where the decoded video is often used for searching in very rich, available databases. Naturally, besides good visual quality, it is also critical to provide good searching performance.

Developments in computer vision have led to the emergence of new forms of visual information representation which are better suited for visual analysis tasks than just pixels. Local visual features are a powerful type of such representations, and have been playing a central role in modern digital image and video applications such as mobile visual search [6], object recognition [7, 8], and scene classification [9]. Such local features describe image characteristics that are distinctive, representative and informative. They are usually obtained by first performing keypoint detection to identify salient image regions and then extracting a descriptor to capture the local characteristics. The Scale Invariant Feature Transform (SIFT) [7, 8] and Speeded-Up Robust Features (SURF) [10, 11] are two major description tools in this context. Following this recent trend, distributed visual analysis systems, for instance, may aggregate a huge amount of data captured from multiple and distributed visual sensors and perform complex visual analysis, targeting to provide services such as augmented reality in sport events, behavior analysis in security systems and mobile

visual search [6, 12, 13]. The latter is a rather mature and increasingly popular application that uses local visual features to retrieve, from a remote server, relevant information for a query image or video. In this context, three main approaches have been considered to meet different constraints when performing feature-based analysis in scenarios involving remote searching. These are the Compress-then-Analyze (CTA), the Analyze-then-Compress (ATC) and the Hybrid-Analyze-Then-Compress (HATC) approaches [14–16]. In the CTA approach, the remote analysis is carried out using visual features extracted from compressed, transmitted and decompressed video content, thus enabling also visualization. However, the compression usually has a detrimental effect in the decoder-extracted visual features which in turn impairs the visual analysis performance. Some works have tried to modify existing standard image and video coding solutions to better preserve the features of interest [17–19]. On the other hand, in the ATC approach, the visual analysis performed at the remote server has to solely rely on a set of compressed visual features extracted and transmitted by the sender. Naturally, such approach has the drawback of not enabling visualization at the server side, which limits the range of applications [20]. For this approach, a significant amount of work has been done with several authors proposing coding schemes to efficiently compress state-of-the-art local visual features such as SIFT and SURF both for images and videos [14, 21, 22]. Also, new visual feature descriptors have been carefully designed, targeting lower bit rate representations [23] such as the so-called binary descriptors [24]. Still in the ATC domain, the recently issued MPEG-CDVS (Compact Descriptors for Visual Search) standard [12, 13] provides description tools to enable interoperability in the context of image searching.

Finally, the HATC approach aims at overcoming the limitations of the two previous paradigms by combining pixel-based and feature-based coding. Considering that visualization and searching are becoming very popular together, the HATC approach has recently attracted attention. In [25, 26], an image coding solution based on SIFT descriptors is proposed already inspired by the technique reported in [27, 28]. SIFT descriptors are extracted from the original image and differentially coded with respect to SIFT descriptors extracted from a poor quality, down sampled and low rate version of the image, that is first conveyed to the decoder. This first image is used to guide the target quality reconstruction, since it should carry enough information about the edges, colors and objects. The decoded descriptors are used to retrieve highly correlated images available in the cloud which shall provide image patches to enable a higher quality image reconstruction. In the context of scene classification and pedestrian detection, a two-part predictive coding architecture is proposed in [9], targeting both the signal (image) and feature fidelities. Related systems are proposed in the contexts of Visual Sensor Networks

(VSN) [16] and augmented reality applications [29]. In [30], a video coding solution is proposed where keypoint information detected on the uncompressed video frames is coded in parallel with regularly coded video, thus not exploiting their synergies. It was experimentally demonstrated that keypoints detected on uncompressed video are effective in reducing the detrimental effects of compression on feature matching performance even if the descriptors themselves are extracted from lossy decoded video [30, 31]; this highlights the importance of using keypoint information extracted from uncompressed data for efficient searching.

In these previous HATC works, pixel and feature-based representations are essentially designed and used independently from each other, meaning that the feature-level data, targeting searching is not exploited to aid the pixel-level coding, targeting visualization, and vice-versa. But this scenario is starting to change. In [20], a hybrid framework for jointly coding the feature descriptors and visual content is proposed, exploiting their interaction. While the feature descriptors are efficiently represented by taking advantage of the structure and motion information in the compressed video stream, the already compressed descriptors can be used to further improve the video compression efficiency by applying feature matching based affine motion compensation.

1.2 Objective and contributions

In modern video applications, the role of the decoded video is much more than filling a screen for visualization. To offer more powerful video enabled applications, it is increasingly more critical not only to visualize the decoded video but also to provide efficient searching capabilities for similar content. Video surveillance and personal communication applications are critical examples of these dual visualization and searching requirements. However, current video coding solutions are strongly biased towards the visualization needs.

In this context, the goal of this research work is to design a novel dual-purpose video coding approach that is more adjusted to the current role of digital video in modern applications targeting both visualization and searching needs by adopting a hybrid coding framework where the usual pixel-based coding approach is combined with a novel feature-based coding approach.

The main contributions resulting from the pursuit of this objective are:

- **Study on the coding performance of video features**

  A comprehensive study has been conducted in order to set up a firm ground for further research on coding visual features extracted from video sequences. This study, published in the conference paper **C.1**, presents a visual feature

coding framework with various coding modes, including intra-frame and inter-frame, with and without decorrelating transforms.

- **Hybrid coding approach based on pixels and visual features**

  This work proposes a video coding architecture that employs a hybrid approach where pixel-based and feature-based coding are jointly used. The periodic k-frames are coded using a standard pixel-based notably HEVC approach and used as reference frames to code the f-frames using a feature-based encoder. Once the k-frames are decoded, frame rate-up conversion is performed to obtain a first coarse estimation of the f-frames. The basic idea to code the f-frames is to refine this coarse estimation by migrating appropriate image patches from the decoded reference frames. This is achieved by establishing correspondences between features/patches in the original f-frames and the already available decoded reference frames. In this way, the quality of the f-frames may be gradually improved by reusing appropriate image patches from the reference frames guided by keypoints extracted from the original data, relying on the fact that video sequences usually exhibit significant temporal redundancy. In addition, since keypoint positions extracted from original uncompressed video data are available for the f-frames at the encoder, the visual searching performance may be boosted compared with the one of decoder extracted keypoints based on lossy decoded video. This is a conceptually refreshing coding approach which tries to conciliate some degree of backward compatibility with HEVC, the most recent video coding standard (through the k-frames) with a new video coding approach targeted at boosting the searching performance (through the f-frames). A preliminary version of this video coding approach based on pixels and visual features resulted in a conference paper **C.2** and an extended version **J.1** has been submitted as journal paper.

- **Joint visualization-searching optimization framework**

  A flexible joint Lagrangian optimization framework is proposed where pixel-based and feature-based processing are combined to find the most appropriate trade-off between the visualization and searching performances. It allows to adjust the balance between the visualization and searching performance up to the extreme cases where one of them is totally dominating, depending on the specific application scenario requirements. This framework offers a synergetic video coding approach between two key user capabilities. This joint optimization strategy constitutes an important part of the submitted journal paper **J.1**.

- **Iterative descriptor matching estimation**

Measuring the visual quality distortion is straightforward because of the availability of the original at the encoder side. However, the situation is very different for the searching capability, as the descriptor matching performance cannot be precisely measured at the encoder as only the decoder has access to the target content database. It is proposed to estimate the descriptor matching performance at the encoder side by mimicking in the best possible way the descriptor matching steps that are performed at the decoder. Such descriptor matching performance estimation enables to formulate a joint Lagrangian optimization to trade-off the rate against the joint visual quality and descriptor matching distortion. This descriptor matching estimation also constitutes an important part of the submitted journal paper **J.1**.

1.3 Outline of the thesis

After this introductory chapter in which the motivation and context within which this work has been carried out as well as its objective and contributions are presented, the remaining content of this thesis is organized as outlined below.

Chapter 2 briefly reviews basic concepts related to digital video signals, their representation and characterization. It also summarizes the main coding tools of the block-based prediction and transform video coding paradigm as well the HEVC coding standard. Furthermore, common full-reference evaluation metrics and some performance results for HEVC are presented and discussed.

Chapter 3 presents a short review of local feature detection and description tools. A coding framework for visual features extracted from video sequences is briefly described, including intra- and inter-frame coding modes.

The objective of the first three chapters is to lay down the ground for the dual-purpose video coding solution presented in the Chapter 4, which starts by describing the walkthrough of the proposed solution and presenting the designed architecture. Afterwards, it describes more in-depth the most novel and technically original coding modules.

Chapter 5 presents extensive experimental results for the assessment of the proposed solution under meaningful test conditions, notably considering not only the joint optimization objectives but also the special cases where the operational points are selected to provide the best performance towards visualization or searching. Finally, Chapter 6 presents the conclusions of this research work and indicates possible directions for further investigations on video coding solutions that aim at address joint visualization and searching needs.

Chapter 2

Digital video coding: an overview

This chapter presents a brief review of digital video signals and their representation. It also includes an outline of the block-based prediction and transform coding paradigm employed in the state-of-the-art video coding technology, its main building blocks consisting of intra- and inter-frame prediction, transformation, quantization and entropy coding. The High Efficiency Video Coding (HEVC) standard is presented as the prominent representative of the aforementioned coding framework. Furthermore, common full-reference evaluation metrics used to drive coding decisions, to assess the quality of the reconstructed signals and to compare different coding solutions are briefly reviewed. Some results obtained by coding a few video sequences using an HEVC encoder are presented and discussed.

2.1 Digital image and video signals

The vast application domains of digital still image and video signals have been producing an ever growing diversity and quantity of data. Although digital image and video signals are not limited to the outcomes of imaging systems that sense the interaction of the visible light with objects of a given scene, this work focus on such types of signals. For example, those produced by digital cameras.

A thorough description of the acquisition process of image and video signals is beyond the scope of this work, but in brief an imaging system aims to sense a continuous 3-D scene, at a fixed instant of time, so that it can be displayed as a matrix $I(x, y)$ of discrete picture elements well-known as pixels [32, 33]. In turn, each pixel is composed of three component samples. In this representation, (x, y) are discrete spatial coordinates in the image plane. A Charge-Coupled Device (CCD) imaging sensor, for example, is constituted as an array of collection sites [34] that collect photoelectrons produced as a result of the incident light. In order to produce color images, CCD imaging sensors use an array of red, blue and green filters with a spatial distribution inspired in the Human Visual System (HVS) [33]. The output

of these spatial filters is somehow combined to produce the Red, Green and Blue (RGB) color components of each pixel in an image. These RGB color components represent the amount of the primary colors at each pixel position. This color representation scheme is based on the HVS tri-stimulus model, a linear combination of RGB components is expected to be able to represent any visible color [35].

Although the RGB color space is often used in the acquisition and displaying stages [5] of the processing pipeline for handling visual information, it is common the use of the YCbCr (or YUV) color space during compression stage as it decouples the most important information for HVS, notably the luminance or luma (Y) component, from the chroma components Cb and Cr. This color representation scheme in combination with an uneven spatial sub-sampling pattern of luma and chroma components results in bandwidth savings with negligible perceptual loss for the HVS. Aspects of conversion between RGB and YCbCr color spaces, quantization and sampling patterns can be found in [36]. Typical sub-sampling formats include the 4:2:2 and 4:2:0 formats [5]. In the 4:2:2 case, for each 4 samples of luma component, there are two samples for each chroma component Cb and Cr. The chroma components having the same vertical resolution as the luma component, and in the horizontal direction half the resolution. In the 4:2:0 format, for every four samples of luma, there is one sample for each of the chroma components. The horizontal and vertical resolutions of the chroma components are half the one of the luma component. The average number of bits per pixel assuming 8 bits per component would be 12 in the case of 4:2:0 format and 16 for 4:2:2.

Digital video signals are constituted by a sequence of images (or frames) sampled at an adequate frequency along the temporal dimension to give the impression of continuous transitions when reproduced. The sampling frequency strongly depends on type of content captured. Television signals, for example, are typically acquired at 25 and 30 frames per second (fps)[5]. Rapid changing scenes might require higher sampling frequency. A large number of image and video signals are produced from natural scenes as the result of an interest in retaining a particular kind of visual information, or to provide more natural communication experiences with the aid of visual information. Therefore, natural image and video signals usually exhibit different textures and well defined structures of the real world. Another class of image and video signals, for example, are those synthetically generated as the ones from special effects in movies and electronic games. Despite the diversity of sources, image and video signals usually exhibit a considerable amount of both spatial and temporal redundancy. Notably, within a frame of a video sequence, the amplitude of the signal often changes smoothly. This is also the case from frame to frame, where the exhibited visual content changes smoothly. Most compression solutions strive to take advantage precisely of these characteristics of image and video signals

in combination with the characteristics of HVS to provide compression efficiency by employing coding tools to reduce both redundancy and irrelevancy.

2.2 Block-based prediction and transform video coding framework

From a broad perspective, video compression can be addressed in a lossy or lossless manner. Lossless compression is suited for applications that impose a perfect reconstruction requirement for the input signal, examples include medical images used to aid diagnose diseases and at production stage in the multimedia and entertainment industry. Nevertheless, as the majority of applications have more relaxed constraints in terms of quality and more strict constraints in terms of bit rate, those applications call for lossy video coding.

Rate-Distortion (RD) theory provides the theoretical foundation for lossy video coding as it describes the trade-off of minimizing the rate for a given acceptable distortion [1, 2, 37]. This problem was first comprehensively addressed by Shannon in the context of representing a continuous random variable with a finite number of bits [38, 39]. The Rate-Distortion function $R(D)$ describes the theoretical bound on the compression efficiency according to an acceptable distortion [37, 40]. Unfortunately, $R(D)$ functions are known only for simple statistical sources [1], although it is useful to keep them in mind when deriving lossy video compression techniques.

Among video coding paradigms, the block-based prediction followed by transform coding is the most successful coding framework, and it is the base of largely deployed video coding standards such as H.264/AVC [5, 41]. It resorts to a set of flexible coding tools matured during decades of research to come up with a very efficient (compressed) description of the input signals. The decision process involved in the course of obtaining this compressed description is a fundamental aspect of image and video encoders. To this end, the employed coding tools are used in the best possible way to achieve the desired RD trade-off.

In the block-based prediction and transform video coding paradigm, the input video frames are usually partitioned into non-overlapping blocks. For each block, a prediction is computed, which can be derived from samples of neighboring decoded blocks of the same frame (intra-frame prediction) or from a list of decoded reference frames (inter-frame prediction). To enable this prediction scheme based on decoded samples, the decoder is embedded in the encoder. It is worth to mention that while the encoder plays an active role in deriving an RD-driven prediction by performing intra-frame estimation as well as motion estimation, the decoder merely follows what is decided in the encoder by carrying out intra- and inter-frame prediction. The

prediction block is subtracted from the current block to form a prediction residue block. Then, the prediction residue block is subjected to a transform step aiming to compact energy and making it more suited for quantization. The quantized transform coefficients along with the prediction mode and motion vectors are entropy coded and transmitted to the decoder so it may follow the decisions made at the encoder side and reconstruct the signal. Figure 2.1 depicts schematically the block-based prediction and transform video coding framework.

The above description has omitted a number of important details, but highlights the main coding tools employed in the mentioned paradigm: intra-frame and motion estimation, intra- and inter-frame prediction, transform, quantization and entropy coding. Below, some of the building block coding tools used in the state-of-the-art video coding technology are briefly described.

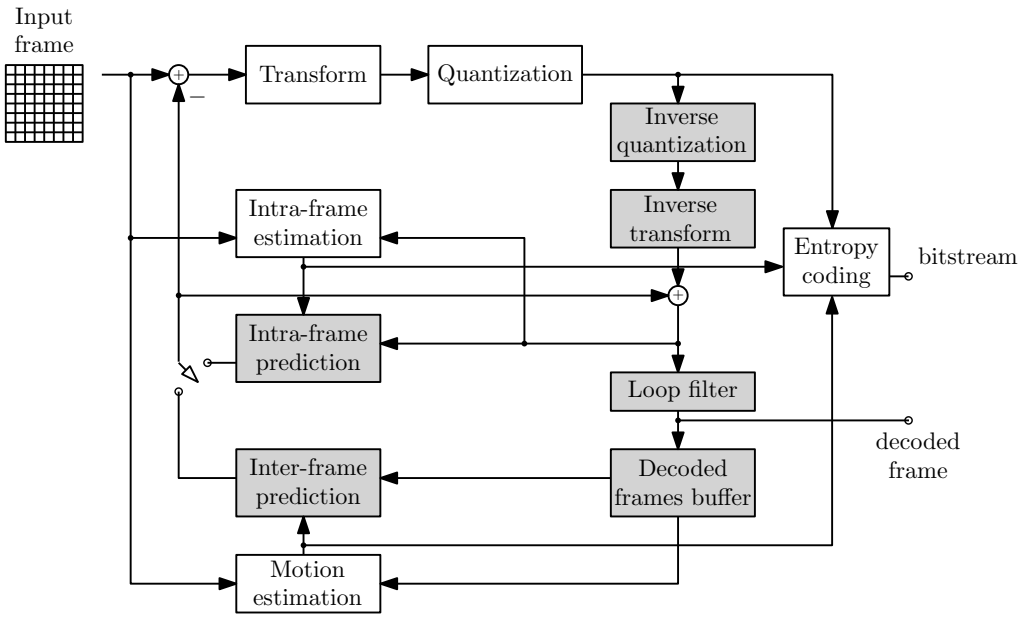


Figure 2.1: Block-based prediction and transform video coding framework. Gray blocks represents the embedded decoder[1].

2.2.1 Prediction

Video sequences usually exhibit a large degree of both spatial and temporal redundancy with fairly structured arrangements of sample values resulting from the well defined structures and texture patterns of natural scenes. Therefore, coding directly the original signal samples without exploiting its redundancy and structure would be inefficient. In this context, using a predictive scheme to remove redundancy and take advantage of the structured nature of image and video signals would be beneficial

[1]Based on the diagram block of standardized video codecs [42].

as the prediction error signal may exhibit less energy and cost less bits to be coded. The prediction error for each image block is derived by subtracting the predicted sample value from each raw sample within the current block. This predicted sample value is derived based on the already coded and decoded samples. The prediction is carried out both at the encoder and decoder, with the encoder playing an active role in deriving an RD-driven prediction and coding the side information required for reproducing the same prediction at the decoder. After that, only the prediction error, that is, the information not present at the receiver side, is conveyed by the encoder to the decoder.

Intra-frame prediction

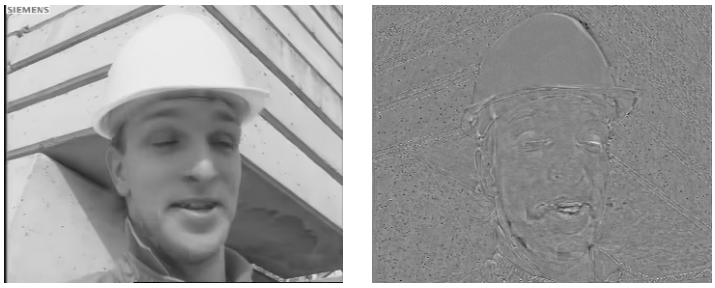
The signal amplitude within an image or a frame of a video sequence often changes smoothly, that is, a particular sample value tends to be close to the values of its near neighbor samples. Naturally, exceptions occur, for example, in object edges and abrupt changes in high frequency regions. Despite that, smooth spatial transition is a good assumption used in intra-frame prediction techniques. The block-by-block coding enables the encoder to use decoded samples from neighboring blocks to predict the samples of the current block to be coded. In the popular JPEG standard [43], for instance, neighboring blocks in the DCT domain are likely to have close DC coefficient values. In view of this, a predictive scheme is used to code the DC coefficients so that only the difference with respect to the DC coefficient of the previous block is coded for each DCT block. Modern intra-frame prediction techniques are quite efficient at taking advantage of pixel correlation to produce a residual signal with much less energy and distribution peaked around zero. Figures 2.2b and 2.2c show, respectively, the intra-predicted frame and the frame difference between the original frame and the intra-predicted frame. This exemplifies the efficiency of current state-of-the-art intra-frame prediction schemes. A good review of intra prediction schemes can be found in [44, 45]. Intra-frame prediction schemes are quite useful in video coding as well, notably for coding the so-called I-frames, which are frames coded without referencing any other frame in the sequence and intend to provide random access and also to offer some resilience to transmission errors. To increase the chance to succeed in obtaining a good prediction for each sample block and to deal with different local sample structures, the intra-frame prediction scheme in use in state-of-the-art video coding solutions such as HEVC [42] can select from various prediction modes available and is able to perform prediction for blocks of different sizes (to some extent). Naturally, as the selected prediction mode must be coded in the bitstream so that the decoder may replicate the prediction, the selected prediction mode must be coded in an efficient way. [46]. Figure 2.3 shows the histograms of the sample amplitude and the residue amplitude for the video

frames shown in Figure 2.2. Details of the used video sequences can be found in the Appendix B.

(a) Original frame.

(b) Intra-predicted frame.

(c) Frame difference between the original and the intra-predicted frame.



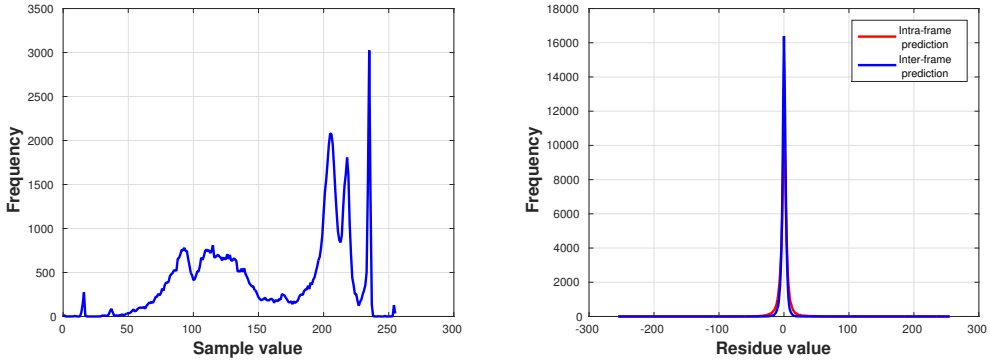
(d) Inter-predicted frame.

(e) Frame difference between the original and the inter-predicted frame.

Figure 2.2: Results of the block-based prediction schemes used in the HEVC for the second frame of the sequence *Foreman*.

Inter-frame prediction

A good assumption often valid in video signals is that the content changes smoothly from frame to frame, resulting in a great amount of temporal redundancy. The goal of inter-frame prediction is to remove as much as possible this temporal redundancy, in order to more efficiently compress the input signal. To this end, a predictive coding scheme is usually employed and instead of coding the frame content directly, one codes just the prediction residue, that is, the novelty of the current frame regarding one or more reference frames. A simple approach would be coding the frame difference without any prior processing. However, to more effectively decorrelate the prediction residue at the cost of an affordable computational complexity, one performs motion-compensated prediction. The current frame to be coded is usually partitioned into sample blocks to better cope with different motions of the objects within the video frames. The prediction residue is computed for each block after performing motion estimation and motion compensation [45, 47–49]. In the motion



(a) Histogram of the original sample amplitude.

(b) Histogram of the residue amplitude.

Figure 2.3: The prediction process produces a residual signal with concentrated distribution making it suited for entropy coding.

estimation, a motion vector is estimated by searching for a matching block in a set of reference decoded frames. A rate-constrained motion estimation is often adopted for obtaining RD profit. Translational motion models are commonly used in this motion estimation process as it is found to be a good balance between the rate to code the motion parameters and the accuracy of motion estimation [50]. In the motion compensation, the matching block in the reference frame is displaced according to the motion vector to generate the motion-compensated prediction. The residue block is obtained computing the difference regarding this motion-compensated prediction. The residue block is supposed to have much less energy and to carry only the novelty that will be added to the prediction block to reconstruct the signal at the decoder side. As the motion vector must be coded in the bitstream so the decoder may replicate the same motion compensation, a great deal of effort must be taken to efficiently signal the motion vectors.

Modern inter-frame prediction algorithms in use in video coding solutions are highly optimized, as they can estimate motion with sub-pixel accuracy and use more than one decoded frame as reference. Also, they are quite flexible, being able to generate motion-compensated prediction for blocks of various sizes in an adaptive fashion. Figures 2.2d and 2.2e show, respectively, the predicted frame derived by motion compensation generated by the HEVC Test Model [51] and the frame difference between the original and the predicted frame.

2.2.2 Transform and quantization

The prediction residue might contain spatial correlation not handled by the prediction step. In view of that, the transform step aims to further decorrelate the

prediction residue. In lossy video coding, the transform coefficients are quantized, therefore the transform should also provide a better representation suited for quantization.

Decorrelating transforms have been the matter of many research works. In the context of block-based video coding, the Discrete Cosine Transform (DCT) plays a fundamental role. It is used in various image and video coding standards such as JPEG [43], H.264/AVC [5, 52] and HEVC [53][2]. The DCT belongs to the class of unitary and separable transforms derived as an approximation to the Karhunen-Loève (KL) transform and has near optimal properties in terms of decorrelation and energy compaction. From a mathematical viewpoint, the transform step decomposes the image block into a set of basis images and associated weights [54]. Due to the energy compaction property of the transform, its coefficients associated to the low frequency base images tend to carry the most significant information, while the high frequency ones tend to be close to zero. It is worthwhile noting that the transformation is reversible, no loss of information occurs. The original image block can be entirely recovered. Next, the transform coefficients are quantized, and depending on the quantization step size, most high frequency coefficients are set to zero since they generally carry only a small amount of the energy of the block content. It is important to notice that there may be some perceptual reasons to quantize more heavily certain frequencies thus increasing the number of zeros and reducing the number of quantization levels and consequently the rate. Most zero valued coefficients are grouped together and only a subset of non-zero coefficients are entropy coded. The decoder is able to reconstruct an approximation of the image block by summing the basis images weighted by the quantized transform coefficients. The quantization introduces irreversible losses as it maps values that have a wide dynamic range to a narrower one, hence a many-to-one mapping. The quantization aims to reduce the amount of bits to represent the DCT coefficients. However, the higher the quantization step size used, the higher the information loss. The quantization step size control allows the encoder to vary the bit rate expenditure and the quality of the reconstructed signal to accommodate different trade-offs.

The use of the DCT in image compression was originally proposed in [55, 56]. In the predictive video coding paradigm, prediction residues (*i.e* image differences) are actually coded. Despite that, the DCT is also widely used for coding prediction residues [57]. An extensive characterization of both intra-frame and inter-frame prediction is found in [45] as well as transforms for coding prediction residues.

Modern video coding standards actually use variable block size transforms that are integer approximations of the DCT to reduce computational complexity [3, 58]. Other discrete transforms are also used in video coding standards. H.264/AVC[5], for

[2]Actually, H.264/AVC and HEVC use integer approximations of the DCT.

example, uses the Hadamard or DCT transforms depending on the type of residual data to be coded. In addition to variable size DCT, HEVC [3] also uses a Discrete Sine Transform (DST).

Figure 2.4 shows the absolute values of the DCT coefficients in logarithm scale for an image block of size 64×64 from the video sequence *Foreman*. Figure 2.5a shows the image block over which it is applied the DCT. Details on the video sequences used throughout this work can be found in the Appendix B. Notice the concentration of the largest coefficient values in the top-left corner of coefficient map as a result of the energy compaction property. Figure 2.5 shows different reconstructions for the block, where the inverse transform is computed considering only a few DCT coefficients, precisely those with the largest absolute values and setting the other coefficients to zero.

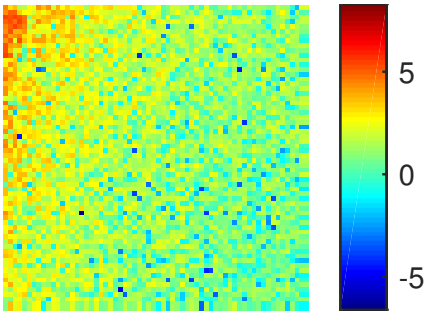
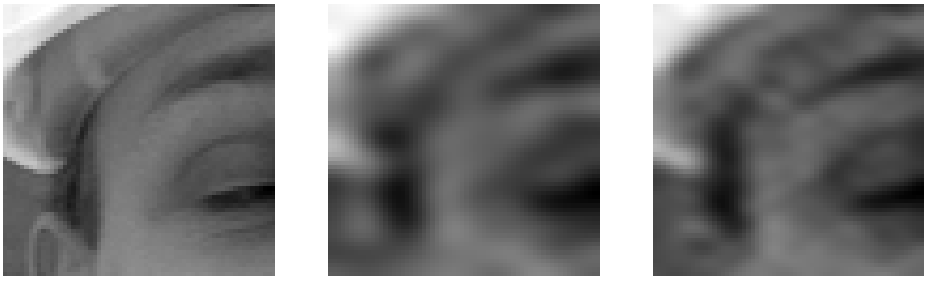


Figure 2.4: Absolute values of the DCT coefficients in logarithm scale.

2.2.3 Entropy coding

The objective of entropy coding is to assign a code for each symbol or sequence of symbols coming from a data source. A quite fundamental and reasonable strategy for achieving compression is to assign shorter codes (codeword) for frequent symbols and longer codes for less frequent symbols, so that the average code length is minimized [59]. For this purpose, a good probabilistic characterization of the symbols from a data source is critical. In addition to minimum average length, it is required to have non-singular codes both for individual symbols and for a sequence of symbols, that is, each symbol (or sequence of symbols) should have a distinctive codeword. Finally, it is desirable that symbols are instantaneously decoded without the need to wait for the end of message. These requirements drove the early developments in entropy coding algorithms. Nowadays, it is also often required efficient context modeling for better prediction of symbol probabilities, adaptive probability models, low memory consumption and parallelism [60, 61].

In video coding applications, the symbols to be coded are the syntax elements describing the video signal. For the block-based prediction and transform video



(a) Original frame.

(b) Reconstruction with the 50 largest absolute values of the DCT coefficients.

(c) Reconstruction with the 100 largest absolute values of the DCT coefficients.

(d) Reconstruction with the 500 largest absolute values of the DCT coefficients.

(e) Reconstruction with the 2.000 largest absolute values of the DCT coefficients.

(f) Reconstruction with the 4.000 largest absolute values of the DCT coefficients.

Figure 2.5: Different reconstruction qualities for a 64×64 samples (luminance) block of the first frame of video sequence *Foreman* resulted by keeping the largest absolute values of the DCT coefficients and setting the others to zero.

coding framework, the syntax elements include block partitioning flags, prediction types, prediction modes, motion vectors and transform coefficients, just to name a few; so that the decoder is able to reconstruct an approximation of the input signal by decoding the bitstream of syntax elements generated by the encoder.

Huffman and Arithmetic Coding (AC) are widely used in image and video coding [59, 62, 63]. Modern video coding solutions such as HEVC [42] use the so-called Context-Adaptive Binary Arithmetic Coding (CABAC) which comprises a binary version of AC with a sophisticated and efficient context modeling in order to achieve high compression ratios [64].

2.3 The HEVC standard

The High Efficiency Video Coding (HEVC) [3, 4, 42] standard jointly developed by ISO/IEC MPEG and ITU-T SG16 VCEG is the state-of-the-art on video coding technology today. It adopts the block-based prediction and transform cod-

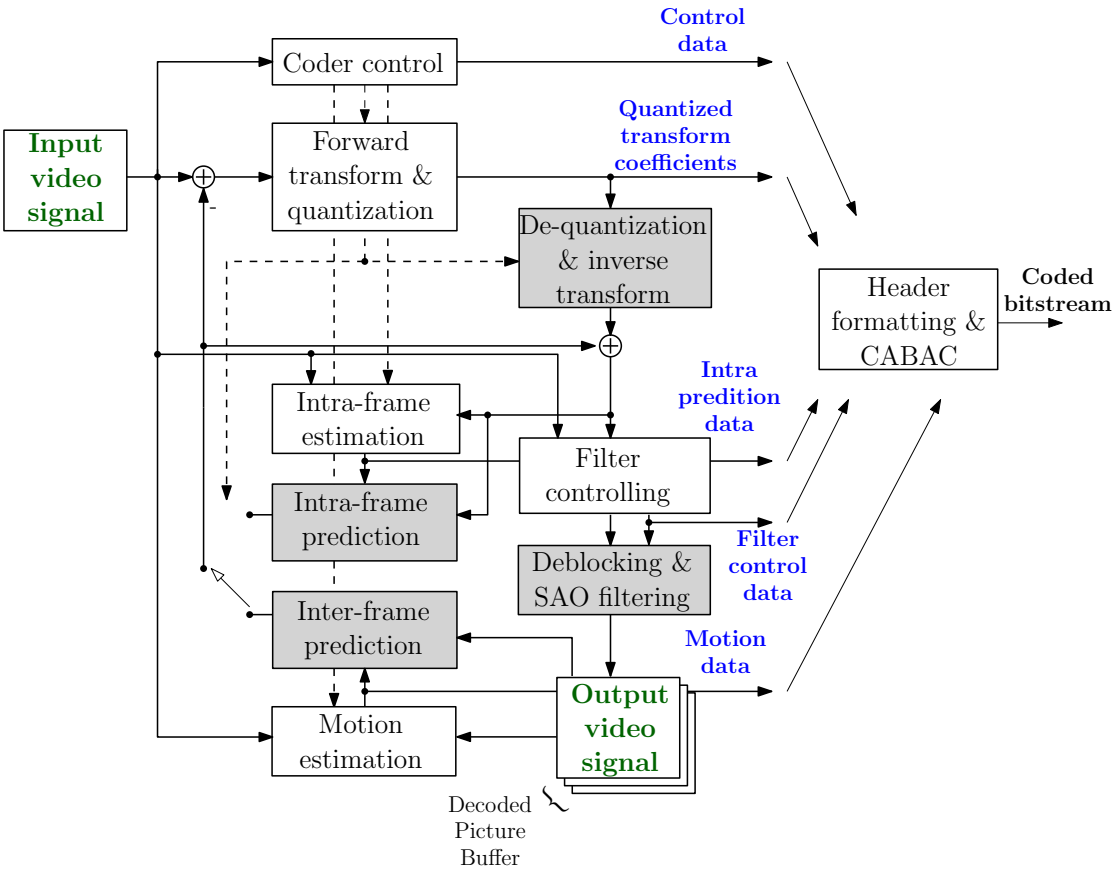


Figure 2.6: Block diagram of a typical encoder for generating an HEVC compliant bitstream. The gray blocks are the embedded decoder blocks[3].

ing paradigm summarized above. HEVC consists of a set of mature coding tools successfully adopted in previous video coding standards and continuously refined during decades of research. Similarly to the previous video coding standard H.264/AVC [5, 41], HEVC was designed to provide efficient video compression solutions for application scenarios ranging from video broadcasting to video storage, internet streaming and video-chat services [53], only to name a few. Moreover, the need to address the increasing resolution in video content and to take advantage of parallel processing architectures have urged the development of a more efficient video compression solution. Following the same approach as previous standards, HEVC standardizes only the bitstream structure, the syntax elements and the decoding process, this way providing interoperability between non-normative encoder and normative decoder implementations. In this context, interoperability is provided solely if an encoder, although non-normative, generates a bitstream in conformance with the standard. Figure 2.6 shows the block diagram of a typical HEVC encoder, for a more in-depth presentation of all coding tools refer to [3]. The several optimization and processing steps made in the course of generating a compliant bitstream

[3]Inspired in the typical HEVC video encoder presented in [3].

are entirely up to the encoder design, this is notably to allow different compromises between compression performance and implementation complexity. The HEVC standard defines a flexible framework by employing nested quadtree structures [65] for better partitioning, prediction and transform coding of the basic processing units defined in the standard. Such approach gives HEVC a high degree of flexibility for adapting its coding tools according to local signal characteristics to obtain a very compact video representation. The coding tools include an intra-frame prediction scheme with 35 prediction modes, inter-frame prediction with sub-pixel accuracy, flexible transform block sizes and efficient Context-Based Binary Arithmetic Coding (CABAC) for entropy coding.

The coding process in HEVC splits each input video frame into non-overlapping square-shaped image blocks. Each one of these image blocks is processed using the basic processing unit defined in the standard, the so-called Coding Tree Unit (CTU). For handling the three color components within the image blocks, the HEVC standard employs a coding structure called Coding Tree Block (CTB) for each color component, thefore a CTU consists of one luma CTB, two chroma CTBs and related syntax elements. The CTB size is configured by the encoder in luma samples and can be of sizes 16×16, 32×32 or 64×64. The corresponding chroma CTBs size obeys the chroma sub-sampling format, for instance, in the common 4:2:0 format, the chroma CTBs size is half the luma CTB in each dimension [3]. By employing a quadtree partitioning structure, the CTB can be coded using multiple Coding blocks (CB) or directly as a single CB [3]. One luma CB and two chroma CBs together form a Coding Unit (CU). Each CU can be classified regarding the employed prediction mode as skipped CU, intra-coded CU and inter-coded CU [65]. Each CB resulting from CTB partitioning can be further split for prediction purposes. In this context, one, two or four Prediction Blocks (PBs) can be derived depending on the prediction mode and CB size, with the allowed PB size ranging from 4×4 to 64×64. The resulting prediction residue associated to the CB is coded resorting to a transform tree, also referred to as Residual Quadtree (RQT). The leafs of this transform tree are the so-called Transform Blocks (TBs). Each CB residual resulting from prediction may be further split for coding purposes into smaller TBs or directly coded as a single TB. The permitted TB sizes are 4×4, 8×8, 16×16 and 32×32. All prediction-related syntax elements, including block partitioning and applied prediction mode, are kept together for the three components in a Prediction Unit (PU). Similarly, all transform-related syntax elements are kept together in a Transform Unit (TU). In turn, the PU and TU form a CU. HEVC also defines high-level syntax concepts, notably for allowing the configuration of parameters and coding features. For example, a slice is a set of consecutive CTUs which can be decoded independently from other slices of the same frame. Three different types of

slices are defined according to the prediction type allowed, namely I-slices, P-slices and B-slices. In I-slices, only intra-frame prediction is allowed to be used, whereas P-slices and B-slices may also be coded resorting to inter-frame prediction. Both uni- and bi-prediction are permitted for inter-frame prediction of B-slices, while for P-slices only uni-prediction is allowed.

In the sequel, the main features of the coding tools adopted in HEVC are highlighted and briefly discussed. One should note that this does not intent to give a thorough description of the HEVC standard. For this purpose the reader can find comprehensive treatments in [3, 53, 65].

2.3.1 Prediction tools

As previously pointed out, predictive techniques play a central role in video coding solutions as they are devised to generate decorrelated residual signals and to provide compression efficiency. To this end, the HEVC standard employs fundamentally two predictive scheme types, notably intra-frame and inter-frame prediction. The decision between intra and inter mode is made at CU level. As for the prediction modes applied for the CBs constituting a CU and resulting from the CTB partitioning, a brief explanation is given in the sequel for intra- and inter-predicted CU.

Intra-frame schemes

The intra-frame prediction in HEVC includes 35 prediction modes, which were devised to propagate decoded reference samples from neighboring TBs into the area covered by a prediction block. The available prediction modes are the DC mode, the Planar mode and 33 Angular modes. Each prediction mode defines a particular propagation rule, thereby enabling approximation of various spatial image structures including near constant, gradient and directional structures [66].

In the sequel, the main aspects for generating the prediction for an intra-predicted CU are summarized.

- **Partitioning rules:** each CB of size $M \times M$ resulting from the CTB partitioning may be predicted using one or four Prediction Blocks (PBs) as schematically depicted in Figure 2.7. The partition into four PBs of $M/2 \times M/2$ is permitted only when the CB size has reached its minimum allowed size. Therefore, the PB can be of size $N \times N$ in units of luma samples, with $N = \{4, 8, 16, 32\}$. Although the PB size is generally defined by the CB size, for intra-predicted CB, the prediction is actually carried out obeying the TB size within the CB [3, 42].

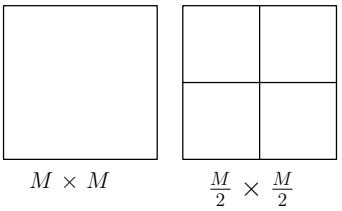


Figure 2.7: Options for CB partitioning into PBs (intra-predicted CU).

- **Reference samples:** the samples located in the neighboring TBs are used for deriving the prediction block. The used TB reference samples may be from the current CB and from neighboring available CBs. A total of $4N+1$ samples around the block to be predicted can be used as reference. Figure 2.8 shows the case when $N = 32$. The samples actually used for generating the prediction block depend on the prediction mode. In order to avoid introducing artificial edges [66], before being used for prediction, the luma reference samples may go through a low pass filtering step. The decision whether or not to apply the filtering depends on the block size and the prediction mode. For the DC, horizontal and vertical modes, the filtering is not applied as well as for blocks with $N = 4$.

- **Prediction modes:** the prediction block for the DC mode is obtained assigning the average of the horizontal and vertical reference samples. A special treatment is given to the predicted luma samples near the reference samples as those predicted samples are smoothed. The prediction for the Planar mode is generated for each sample within the prediction block by computing a weighted average using four reference samples. The prediction using Angular modes is more elaborated, the complicating factor is to determine the reference samples to be used for computing the predicted sample. In view of this, HEVC maps the reference samples into an 1-D array, each Angular mode defining a different mapping. Once the 1-D array is available, the prediction for each sample within the prediction block is computed interpolating two reference samples of this 1-D array. The interpolation is carried out with 1/32 sample accuracy. The Angular modes are schematically depicted in Figure 2.8 for the case where $N = 32$.

- **Prediction mode signaling:** two alternative options are available in HEVC for signaling the chosen prediction mode for each PB. The first option is to code an index indicating one of the three prediction modes available in a list of Most Probable Modes (MPM). Alternatively, a fixed 4-bit code is coded to indicate one of the 33 modes not included in the MPM. The MPM list is derived for each block to be predicted and it is conditioned to the availability

of neighboring blocks and its prediction modes. The two alternatives are coded using CABAC in bypass mode.

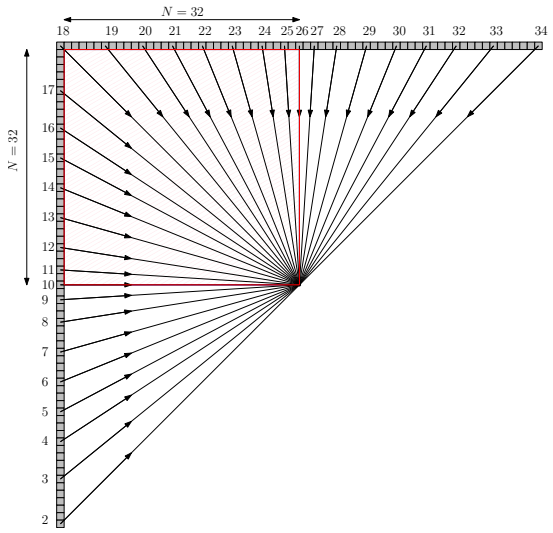


Figure 2.8: Angular prediction using decoded reference samples from neighboring TBs. The horizontal and vertical little grey blocks represent reference samples. The highlighted area in red represents the sample block to be predicted.

Inter-frame schemes

The underpinning idea for inter-frame prediction in HEVC is to derive a residual signal carrying only the novelty of the frame to be coded regarding sample blocks of decoded reference frames. This is deemed to be a fundamental idea for efficient representation of video signals. HEVC is equipped to generate a 'mosaic' of motion-compensated PBs by exploiting for prediction two lists of reference frames, the so-called list 0 and list 1, which are constructed from the Decoded Picture Buffer (DPB). To better exploit the reference lists, in addition to variable size PBs, HEVC may be given the choice between *uni-* and *bi-prediction* modes (depending on the slice type). To obtain a prediction for the sample block within the domain of a PB, an encoder may perform motion estimation by searching for a block match in the list of reference frames so that an RD criterion is minimized. The resulting motion vectors (vertical and horizontal displacement values) as well as references to frames in the reference lists are coded and sent to the decoder so it can replicate the motion-compensated prediction.

In the sequel, the main aspects for generating the prediction for an inter-predicted CU are summarized.

- **Partitioning rules:** in order to generate a prediction, each CB of size $M \times M$ resulting from the CTB partitioning may be further partitioned into one, two

or four PBs. The allowed shapes for partitioning a CB into PBs are shown in Figure 2.9. The partition option $M/2 \times M/2$ is only allowed when the CB has reached its minimum size. The partition shapes in the second row of Figure 2.9 are named Asymmetric Motion Partitions (AMP) and are included in the design as they may provide an efficient representation when a foreground object partially overlaps the background within the domain of a CB [42].

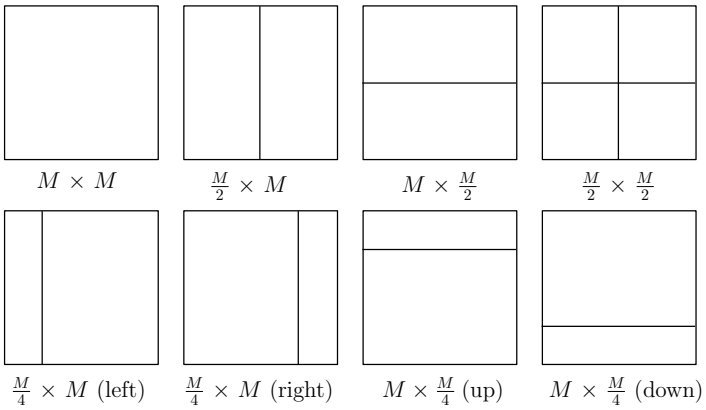


Figure 2.9: Options for CB partitioning into PBs (inter-predicted CU).

- **Sample interpolation:** for improved motion-compensated prediction, the reference sample block used for prediction may be displaced with non-integer sample accuracy. In this case, additional fractional samples must be interpolated between samples in the integer sampling grid. The interpolated fractional samples can be generated with accuracy of one quarter of sample for luma component. As for the chroma components, the accuracy depends on the sampling format. For the common 4:2:0 format, the interpolation is carried out with accuracy of one eighth of a sample. Two filters are defined for interpolating the luma samples at fractional sample positions, the fractional samples at half-sample positions are obtained with an 8-tap filter, whereas fractional samples at quarter-sample position are obtained with a 7-tap filter. For chroma fractional sample interpolation, a set of 4-tap filters are defined for the usual 4:2:0 sub-sampling format. The prediction samples may go through scaling and offsetting operations in case of weighted prediction as well as rounding, bit-shifting and clipping operations in order to keep the prediction samples in the original bit-depth. To this end, HEVC provides a simplified design resulting in a greater flexibility and decreased rounding errors regarding the previous standard H.264/AVC. Further details on this subject can be found in [3, 42].

- **Inter-frame prediction modes:** the motion-compensated prediction can be derived resorting to one or a combination of two reference blocks, these modes

are named *uni-* and *bi-prediction* schemes, respectively. The allowed modes depend on the slice type. For B-slices, both uni- and bi-prediction are allowed. In this case, the two reference lists may be used for generating the prediction block. On the other hand, only uni-prediction is allowed for P-slices, in which case only reference list 0 is used. For improved prediction performance, a weighting factor and a scale offset may be applied to the predicted samples. This weighted prediction is defined for both uni- and bi-prediction schemes. Notably, for the bi-prediction case, weighted prediction is used to combine two reference blocks.

- **Motion vector information representation:** two efficient representations of motion information are defined in the HEVC standard, notably the *merge mode* and a predictive motion vector coding scheme named *Advanced Motion Vector Prediction* (AMVP). Rather than explicitly signaling the applicable motion vectors and related information (notably, reference frames indices and used reference lists), in the merge mode, the motion information used for obtaining the motion-compensated prediction is indicated by an index which identifies one out of several candidate motion vectors in a list. The list of candidate motion vectors is built in an adaptive fashion from both spatially and temporally neighboring blocks. The skip mode is regarded as a particular case of the merge mode. A predictive scheme is employed when the merge mode is not chosen for representing the motion information. A candidate list of two predictors is built from spatially and temporally neighboring blocks, similarly as for the merge mode, and only the motion vector difference regarding one of the reference predictor needs to be indicated. The reason to maintain a reduced list of candidates in the predictive motion vector coding scheme is to keep an affordable computational complexity for motion estimation in the encoder [42, 67].

2.3.2 Residual coding

Before entropy coding, the most common approach in HEVC for processing the residual blocks resulting from the prediction step is to apply a decorrelating transform and a quantization process for reducing the number of bits required to represent the selected transform coefficients. Alternatively, it is also possible to skip the transform and quantization steps. Because of the energy compaction property of the transform and the quantization applied, the non-zero coefficients tend to cluster together. To exploit this fact, HEVC defines scanning patterns for efficient coding of the transform coefficients.

Used transforms

The HEVC standard defines two transforms for decorrelating the residual blocks, notably the DCT and the DST. To achieve compression efficiency, HEVC employs a Residual Quadtree (RQT) structure for coding each CB resulting from the CTB partitioning process. The leafs of this RQT structure are the so-called Transform Blocks (TBs). As the roof of this transform tree is the CB, in the case of inter-predicted CU, the transform blocks are allowed to encompass more then one PB if it is found to be beneficial for compression efficiency. The actual transforms defined in the HEVC standard are finite precision approximations for both the DCT and DST. Because the defined transforms are separable, the 2-D transform is computed by applying a 1-D transform along one direction and then repeating in the other direction.

- **DCT:** in its design, the DCT integer approximation has taken into account precision, closeness to orthogonality and control of the dynamic range in the transform computation. The allowed TB sizes are $\{4\times4, 8\times8, 16\times16, 32\times32\}$. For simplicity, a single matrix of size 32×32 is defined and the matrices for other TB sizes are obtained by sub-sampling the 32×32 defined matrix.

- **DST:** the 4×4 luma residual blocks obtained resorting to intra-frame prediction are treated differently. For this case, an integer approximation of the DST is applied instead of the DCT. This special treatment provides roughly 1% of bit rate savings in intra-frame prediction coding [3, 68].

Quantization

The HEVC standard allows 52 step sizes for uniform quantization of the transform coefficients. These quantization step sizes are indexed by the Quantization Parameter (QP), which can assume values from 0 to 51. The quantization step size and the QP are related by $Q_{step}(\mathrm{QP}) = 2^{\frac{\mathrm{QP}-4}{6}}$. Therefore, the quantization step size is defined within the interval $0.630 <= Q_{step} <= 228.1$. In this context, the higher the QP, the higher the quantization step size and the information loss incurred in the quantization. For the integer implementation of the procedure to obtain the quantization level for a given coefficient value (quantization) and the coefficient value for a given quantization level (de-quantization), several scaling, bit-shifting and rounding operations are performed. For further details on these topics, refer to [69]. In addition to an even quantization scheme, HEVC also allows for the use of a weight matrix for adapting the quantization step size depending on the coefficient frequency. Both predefined and custom matrices are allowed. The use of

a weight matrix may be particularly useful for the design of quantization schemes which consider HVS characteristics.

Scanning patterns

For exploiting the sparsity of the transform coefficients due to the adopted decorrelating tools and quantization process, also targeting compression efficiency, the HEVC standard defines an elaborated representation for the transform coefficients. Three scanning patterns are first defined for 4×4 TBs, notably the diagonal, the vertical and horizontal patterns, as schematically shown in Figure 2.10. For larger blocks sizes, the TBs are split into non-overlapping 4×4 TB sub-blocks and the same pattern applied for scanning the coefficients within the 4×4 sub-blocks is also applied for scanning the sub-blocks. For inter-predicted TBs, only the diagonal scanning is allowed for all TB sizes. On the other hand, all three scanning patterns are permitted for intra-predicted TBs of sizes 4×4 and 8×8. The applicable scanning depends on the prediction direction used. For blocks 16×16 and 32×32, only the diagonal is allowed.

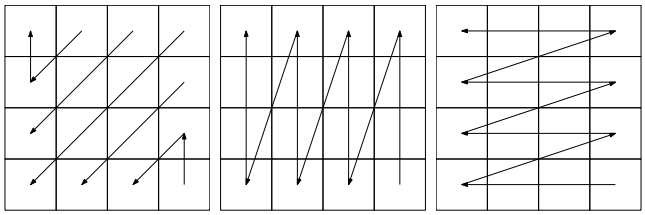


Figure 2.10: Transform coefficients scanning patterns.

In order to generate all syntax elements for coding the quantized transform coefficients within the transform block, the chosen scanning pattern is repeatedly applied. In each scanning, a particular piece of information is generated, namely *significance map*, *level greater than 1*, *level greater than 2*, *coefficient sign* and *remaining absolute level*. Before sending those pieces of information, the position of the last significant coefficient relative to the DC coefficient position (top-left corner of the block) is coded. Also, for each 4×4 TB sub-blocks, a flag is coded to indicate the presence of non-zero coefficients.

2.3.3 Context-adaptive binary arithmetic coding

The Context-Adaptive Binary Arithmetic Coding (CABAC) is the adopted entropy coding technique in the HEVC standard [64]. It was first introduced in the H.264/AVC standard [61]. Since then, it has received improvements mainly to reduce data dependencies due to the large number of contexts and to increase its

throughput by means of parallel-processing [64]. The several syntax elements resulting from describing the input video signal using the defined coding tools are all coded with CABAC. From a schematic point of view, the processing steps for generating the coded bitstream from the syntax elements are depicted in Figure 2.11 and briefly discussed in the sequel.

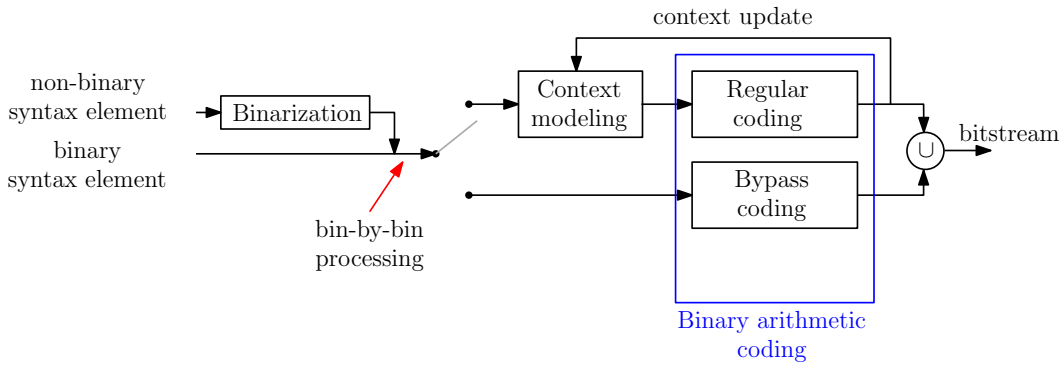


Figure 2.11: Main processing steps for entropy coding the syntax elements using CABAC.

- **Binarization:** all non-binary syntax elements are binarized using one of the defined binarization methods, which includes unary, truncated unary, Exp-Golomb and fixed length. The applied method mainly depends on the syntax elements but may also depend on the value of previously processed elements and slice parameters. Each element of the resulting binary string is called a *bin*. The adopted binarization methods assure the binary string is prefix-free for each syntax element.

- **Regular and bypass coding modes:** a fixed uniform probability model is assumed for the *bins* in the bypass coding mode, thereby no context modeling is applied. The more extensive usage of the bypass mode has contributed for the higher throughput of HEVC regarding H.264/AVC [3]. In the regular coding mode, the use of sophisticated context modeling provides improved compression efficiency. For each *bin*, a probability model associated to the selected context model is used for (binary) arithmetic coding. After each coding and decoding run, the probability models may be updated to better fit the probability of the occurring symbols. Several parameters are considered for context modeling, including the syntax element type, *bin* position, partitioning depth of the coding tree and neighboring information, only to name a few.

2.3.4 In-loop filtering

The block-based processing adopted in HEVC has the drawback of introducing visible blocking artifacts. This unpleasant characteristic originates from discontinuities introduced by operating prediction and transform on a block basis [70]. As in H.264/AVC, HEVC defines the operation of a deblocking filter over the reconstructed signal for reducing blocking structures. Another filtering operation first introduced in HEVC is intended to reduce undesirable artifacts that could became noticeable because of the use of large transform blocks and longer interpolation filters [71], the so-called Sample Adaptive Offset (SAO) filtering. The two filtering steps are applied over the reconstructed signal, SAO after the deblocking filtering. The filtered frame is stored in the decoded picture buffer for displaying and possibly to be used for inter-frame prediction.

- **Deblocking filter:** the filtering operation is applied only to the samples at the boundary of the prediction blocks and transform blocks, with the minimum block size restricted to 8×8. The filtering operation is an adaptive process in which several parameters are considered in order to decide whether or not the filtering is to be applied as well as to decide the filtering strength. In summary, two thresholds are derived from the QP values of neighboring blocks and the filtering strength b_s decided at encoder. The local decision to apply the filtering is drawn by thresholding derivative measures over the boundary samples.

- **Sample adaptive offset (SAO):** the operation mode for applying the SAO filtering is decided by the encoder on a CTB basis among edge offset mode, band offset mode and not applying it. The SAO filtering operation conditionally modifies the decoded samples based on offset values sent by the encoder. In the edge offset mode, the filtering operation depends on the relation of a sample with its neighborhood, whereas the band offset mode solely considers the sample intensity. An interesting aspect of SAO filtering is that in order to determine offset values and applicable offset mode, an RD optimization may be carried out at the encoder side.

2.4 Full-reference video quality objective metrics

Objective metrics to evaluate the quality of reconstructed image and video signals are of prime importance in lossy image and video compression, mainly for driving encoder decisions in the RD optimization, but also for ranking competing compression algorithms in comparative evaluations. Given the original image or video and the decoded image or video, one may want to measure how 'distant' the decoded

signal is from the original reference signal. In particular, for the usual displaying applications such quality metric should agree with human perception.

Peak signal-to-noise ratio

Among the visual evaluation metrics used in the context of image and video compression, the peak signal-to-noise ratio (PSNR) is widely adopted due to its appealing simplicity [72]. It is based on the distortion measure Mean Squared Error (MSE), which accounts the average difference between the samples of the decoded signal and those of the original signal. More precisely, it is given by:

$$\mathrm{MSE} = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} (I(x_i, y_j) - \hat{I}(x_i, y_j))^2 \tag{2.1}$$

W denotes the width and H the height in number of samples. I denotes the original video frame and $\hat{I}$ the reconstructed video frame

The PSNR aims at measuring the resemblance between the samples of the decoded signal and those of the given reference original signal. Therefore, in principle, the bigger the better. Putting aside the discussion about the suitability of PSNR for assessing the visual quality in the context of image and video compression [72, 73], the PSNR is adopted in this work. In precise terms, the PSNR is defined as follows:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{2^b - 1}{\mathrm{MSE}} \right) \tag{2.2}$$

where b is the bit depth, which for the video sequences used in this work is equal to 8. The PSNR is often averaged over all frames to provide a single PSNR value for the whole video sequence.

Regarding the RD optimization process, in addition to the common MSE, an encoder may also use other difference measures such as the Sum of Squared Differences (SSD) and the Sum of Absolute Differences (SAD).

Structural similarity measurement

The Structural Similarity Measurement (SSIM) index is an alternative complementary metric to the PSNR [74]. Relying on the assumption that human vision is adapted to extract structural information from the viewed scene, the SSIM is proposed as an objective metric to predict the perceived visual quality based on the deformation of the structural information, luminance and contrast measures [74]. The SSIM index is computed locally on a block basis with the overall SSIM index being obtained by averaging over all blocks within a frame. Let $\mathbf{b}$ be the vectorized samples of an image block for which one want to computed the SSIM index

regarding the original vector $\mathbf{a}$ of original samples.

In the general case, the SSIM index is given by:

$$\mathrm{SSIM}(\mathbf{a},\mathbf{b}) = \left(\frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1}\right)^{\alpha} \left(\frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2}\right)^{\beta} \left(\frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3}\right)^{\gamma} \tag{2.3}$$

where α, β and γ are parameters for adjusting the contribution of the components and C_1, C_2 and C_3 are constants included to avoid instability. μ_a and μ_b are the mean intensity of $\mathbf{a}$ and $\mathbf{b}$, respectively. Accordingly, σ_a and σ_b are the standard deviations and σ_{xy} is the correlation coefficient. In the particular case for which $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$, the SSIM is given by:

$$\mathrm{SSIM}(\mathbf{a},\mathbf{b}) = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)} \tag{2.4}$$

Bjontegaard Deltas

The Bjontegaard Deltas [75], namely the Bjontegaard Delta PSNR (BD-PSNR) and the Bjontegaard Delta Rate (BD-Rate), are quite useful measures for comparing different image and video coding solutions. Given two sets of (Rate, PSNR) points corresponding to RD operational points of two competing video coding solutions, the BD-PSNR accounts the average PSNR difference (in dB) at the same bit rate between two coding solutions, whereas the BD-Rate accounts the average bit rate difference for the same quality (in percentage). As an example, suppose the BD-Rate between two RD curves resulted from the video coding solutions $\mathbb{A}$ and $\mathbb{B}$ is -3%, with coding solution $\mathbb{A}$ playing the role of the benchmark. This means that the solution $\mathbb{B}$ delivers the same reconstruction quality (PSNR) while saving on average 3% of the bit rate. In the case of BD-PSNR, if the BD-PSNR between coding solutions $\mathbb{A}$ and $\mathbb{B}$ is $+0.5$ dB, this means that on average the coding solution $\mathbb{B}$ delivers 0.5 dB more quality than $\mathbb{B}$ at same bit rate. The Bjontegaard deltas are used in this work for performing comparative evaluation of the proposed coding solution regarding state-of-the-art benchmarks.

2.5 HEVC compression performance

Since HEVC will be the main reference for benchmarking the proposed video coding solution, this section presents a set of experimental results aiming to show the compression performance of the HEVC, notably in comparison with its predecessor H.264/AVC under various coding setups. For this purpose, the HEVC Test Model (HM) [51] version 16.3 and the H.264/AVC reference software JM [76] ver-

sion 19.0 are used. Although the HEVC tools were designed focusing on high spatial resolution video signals such as 4K, the video signals and test conditions used in this chapter and throughout this work reflect the video coding scenario addressed in this research work. For instance, video-based surveillance systems often require low-delay processing and deal with low spatial resolution video signals.

The test conditions and materials used for this experiment are as follows.

- Three video sequences were used, namely *Hall*, *Container* and *Paris*. All sequences are in CIF spatial resolution at 30 Hz and 10 seconds long (300 frames). More information on the used video sequences can be found in the Appendix B. Only the first half of the sequences is compressed.

- HEVC Main profile with four prediction structures were experimented: All Intra, IBI, IBBBI and IBBBBBBBI, where (IB...I) means that in each Group Of Pictures (GOP) the first and last frames are I-frames and the intermediate frames are B-frames in a hierarchical coding structure. The prediction structures and parameter settings for the H.264/AVC were set accordingly as those of HEVC whenever possible. The High Profile was used for H.264/AVC.

- For both HEVC and H.264/AVC, the experimented QP values for the I-frames were 25, 30, 34, 37, 40 and 45. To implement some cascading, the QP values used for B-frames were incremented by 1 regarding the QP values of I-frames.

Figures 2.12, 2.13 and 2.14 show the RD performance using the adopted setups. The charts show the PSNR of the luminance component averaged over all frames as a function of the bit rate. As could be expected, HEVC consistently outperforms H.264/AVC in all coding setups, this is mainly due to its flexible coding tools discussed previously. Notably, the use of a larger basic processing unit than that of H.264/AVC in combination with the nested quadtree structures for better partitioning, prediction and transform coding. Table 2.1 presents the Bjontegaard deltas for HEVC with regard to H.264/AVC. One might notice that the bit rate savings provided by HEVC regarding H.264/AVC are well below the 50% usually provided by the HEVC standard. This is so because of the low spatial resolution video signals used in those experiments, for which, unlike in the case of high spatial resolution video sequences, the Coding Block (CB) size of H.264 (this means 16×16) seems to be quite adequate.

Regarding the coding setups, both HEVC and H.264/AVC behave in the same way. Naturally, All Intra coding setup requires much higher bit rate for achieving the same quality level of those coding setups exploiting inter-frame redundancy. This is because in the All Intra coding setup, all frames are coded independently, only intra-frame redundancy is exploited. Although it is not a good approach for

compression efficiency, it still might find application in scenarios requiring random access. For instance, at editing stage in video production industry. In the case of the coding setups exploiting inter-frame redundancy, the encoder is able to represent more efficiently the video content by resorting to decoded reference frames for deriving motion-compensated prediction, this way only the novelty needs to be coded. This approach provides significant bit rate savings, especially for the static background areas present in the tested video sequences. One can notice that as the number of frames coded exploiting inter-frame redundancy increases, the compression performance also increases, although with smaller incremental gains. The use of larger GOP increases the relative distance between (some) frames to be coded and the reference frames used for deriving their prediction, what in turn, may impair the prediction efficiency, despite the use of a hierarchical prediction structure.

HEVC Complexity

According to available studies on complexity analysis of the HEVC [4, 77], an HEVC encoder is expected to have a significantly higher complexity relative to its predecessor H.264/AVC. This is due to the increased coding flexibility in HEVC adopting large block sizes, nested quadtree structures for segmentation, prediction and transform coding for the input video signal. This set of flexible coding tools implies that an encoder has to assess various coding strategies and parameter settings in the course of generating a compliant bitstream for leveraging the full compression efficiency of the HEVC standard. Although not intended to be an optimized implementation, the HEVC Test Model is put under scrutiny in [4]. The executing time of several coding tools is presented revealing that the most time consuming functions are those related to RD optimization and prediction steps. Regarding decoder complexity, the authors argue that it does not appear to be significantly higher than that of H.264/AVC [4].

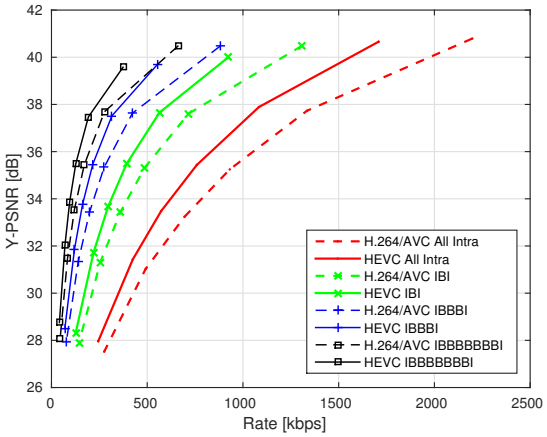


Figure 2.12: HEVC and H.264/AVC compression performance for *Hall*.

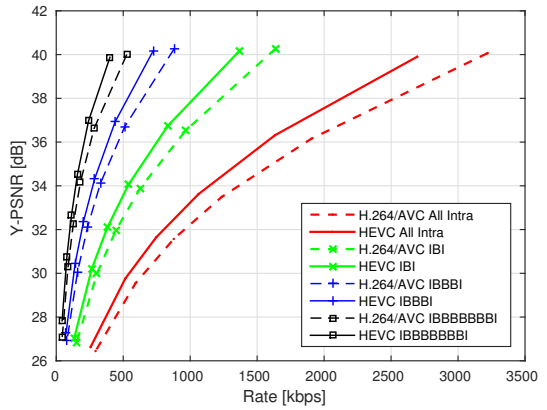


Figure 2.13: HEVC and H.264/AVC compression performance for *Container*.

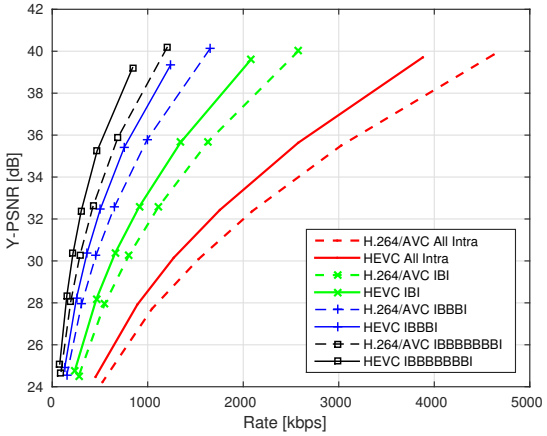


Figure 2.14: HEVC and H.264/AVC compression performance for *Paris*.

Table 2.1: HEVC Bjontegaard deltas regarding H.264/AVC for the four tested setups.

	All Intra	**IBI**	**IBBBI**	**IBBBBBBBI**	
BD-PSNR	1.41	1.26	1.18	1.12	*Hall*
BD-RATE	-19.89%	-20.41%	-21.85%	-23.93%	
BD-PSNR	0.98	1.00	1.03	1.10	*Container*
BD-RATE	-16.03%	-16.26%	-16.71%	-18.41%	
BD-PSNR	1.32	1.38	1.50	1.66	*Paris*
BD-RATE	-16.88%	-18.13%	-20.62%	-24.41%	

2.6 Final remarks

This chapter has briefly reviewed the digital video signals and their representation as well as the main coding tools employed in the state-of-the-art block-based prediction and transform video coding paradigm. Furthermore, it has reviewed the HEVC standard over which the dual-purpose video coding solution proposed in Chapter 4 relies on for pixel-based coding approach. The next chapter briefly reviews local feature representation for visual content and coding schemes devised to code visual features extracted from video sequences, the objective is to lay down the ground for the feature-based coding approach as presented in Chapter 4.

This page is intentionally left blank.

Chapter 3

Local feature representation for visual content

This chapter starts with a brief introduction to local visual features. In particular, it reviews the two main steps for obtaining a set of local visual features, notably feature detection and feature descriptor extraction. Before going any further, the features of interest in this work are localized image features, also referred to in the literature as salient points, interest points or keypoints. Each feature is described by a descriptor vector which is extracted from a local neighborhood around the keypoint location. The seminal Scale-Invariant Feature Transform (SIFT) will be used in this work to characterize the input video frames in terms of local visual features, therefore, this chapter gives an overview of its feature detection and descriptor extraction steps. Moreover, this chapter also presents a review of coding schemes devised to code visual features extracted from video sequences. The purpose of this introduction to visual features is to lay down the ground for the video coding solution proposed in Chapter 4 which relies on visual features.

3.1 Introduction

Developments in the field of computer vision have led to the emergence of visual information representations which are better suited for visual analysis tasks than just pixels. Local visual features are a powerful type of such representations which are able to efficiently perform a number of tasks including, but not limited to, image and video search and retrieval [6, 13], object recognition [7, 11], scene classification [9, 78] and automatic panoramic image stitching [79]. Local feature detectors are designed to produce stable and repeatable responses at image patterns which differ from their nearby neighborhood [80]. The objective is to describe image characteristics that are distinctive, informative and ultimately good for establishing feature correspon-

dences. As a matter of fact, the ability to identify stable image locations and to establish correspondences between them is the starting point for several interesting applications as those listed above.

The review given in what follows does not intent to be exhaustive, even because there is a whole body of research dedicated to images features with decades of developments. A rather extensive survey on local feature detectors can be found in [80, 81]

3.2 Local feature detection

A common issue in many computer vision problems is to establish correspondences between regions of two or more images of the same scene or object [35]. For this purpose, one may need first to identify repeatable and stable image locations for features representation. In this regard, the most desired property for local visual features is *repeatability*. More precisely, this is the ability for the same features to be repeatedly detected on two or more images depicting a common content, although acquired or processed differently. Local visual feature detectors are often designed to give rise to repeatable local features, invariant to image changes such as rotation, scaling and affine. Additionally, they should have properties such as *distinctiveness*, that is, they should show enough variation so that they can be properly distinguished and matched; *locality* to cope with occlusions and to allow simple modeling of geometric and photometric deformations; *quantity*, meaning that the spatial distribution and quantity of local features should reflect the information content within a given image so that a reasonable number of interest region correspondences may be produced at the descriptor matching step; and *efficiency* to fulfill time and computational constraints [80].

A considerable number of local feature detectors have been proposed in the literature, notably for detecting image structures such as junctions, corners [82], blobs [11] and edges, naturally, fulfilling different requirements of the target application scenarios [8, 11, 13, 83] and having different motivating ideas. To name a few, the Harris corner and edge detector was proposed aiming to represent natural images containing roads, buildings, trees and buses [82]. The Harris detector has inspired many works on features detection, notably towards scale and affine invariance such as the Harris-Laplace and the Harris-Affine detectors [81, 84]. The Scale-Invariant Feature Transform (SIFT) has considered in its design invariance to rotation and scale changes as well as robustness to viewpoint and illumination changes. It has shown its strengths in object recognition [7, 8]. In addition to rotation and scale invariance, the Speeded-Up Robust Features (SURF) detector has targeted fast detection and shown excellent results for camera calibration and object

recognition [10, 11]. Targeting real-time applications, the Features From Accelerated Segment Test (FAST) detects corners by evaluating the sample amplitude of a candidate position on the image plane with respect to samples of a circular pattern around it. A similar underlying idea has been used in the Binary Robust Invariant Scalable Keypoints (BRISK) feature detector [85, 86].

In general, keypoints are detected searching for maxima or minima in some intermediate representation of the image. For instance, the Harris [82] corner and edge detector is based on the analysis of the eigenvalues λ_1, λ_2 of the second moment matrix $M \in \mathbb{R}^{2\times2}$, which in turn is derived from the weighted sum of squared difference surface (or autocorrelation function) by using a first-order Taylor approximation [35]. The second moment matrix (also known as structure tensor) is given by:

$$M(x,y) = \begin{bmatrix} w * I_x^2(x,y) & w * I_x(x,y)I_y(x,y) \\ w * I_x(x,y)I_y(x,y) & w * I_y^2(x,y) \end{bmatrix} \tag{3.1}$$

where $I_x(x,y) = \partial I(x,y)/\partial x$ is the image derivative in the x direction, $I_y = \partial I(x,y)/\partial y$ is the image derivative in the y direction, w is a Gaussian window and $*$ is the convolution operation. The second moment matrix describes the gradient information around a neighborhood of a point (x,y) and provides a way to detect corners and edges in a rotationally invariant manner.

Harris has stated that the eigenvalues of $M(x,y)$ will be proportional to the principal curvatures of the local autocorrelation function [82]. An analysis of the eigenvalues λ_1 and λ_2 allows to infer on the shape of the autocorrelation function and on the characteristics of the underlying windowed image, notably: *a*) if both eigenvalues are small, the autocorrelation function is flat, what in turn indicates a near constant region; *b*) if one eigenvalue is large and other is small, the autocorrelation function has a ridge shape, thus indicating the presence of an edge; and *c*) if both eigenvalues are large, then the autocorrelation has a clear peak, that indicates the presence of a corner.

Figure 3.1 exemplifies the three cases listed above, namely a corner (indicated by 1), a near constant region (indicated by 2) and an edge (indicated by 3). The second-moment matrix for each case is given in Equation 3.2 below:

$$M_1 = \begin{bmatrix} 243.59 & 98.97 \\ 98.97 & 201.43 \end{bmatrix}; M_2 = \begin{bmatrix} 8.14 & 4.63 \\ 4.63 & 16.30 \end{bmatrix}; M_3 = \begin{bmatrix} 881.34 & 13.36 \\ 13.36 & 2.00 \end{bmatrix} \tag{3.2}$$

where the eigenvalues for M_1 are 121.31 and 323.71; for M_2 are 6.04 and 18.40; and for M_3 are 1.80 and 881.54. In fact, one can notice the agreement with the description above.

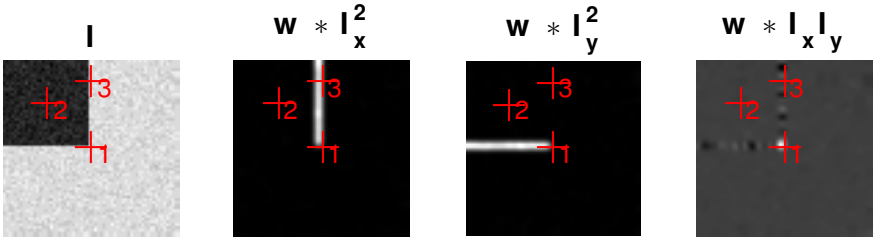


Figure 3.1: Examples of corner, edge and near constant regions. A Gaussian window of unitary standard deviation was used for computing the components of the second moment matrix.

Harris has also proposed a measure of corner and edge quality (strength) based on the trace and determinant of the second-moment matrix:

$$C = \mathrm{Det}(M) - k \cdot \mathrm{Tr}^2(M) \tag{3.3}$$

where k is typically set to 0.04 [80]. The value of C will be negative in the presence of an edge, small for near constant regions and positive for corners.

The Harris corner and edge detector has inspired several improvements and new works, notably towards scale-invariance with the aid of scale-space theory [87, 88] and affine-invariance [80, 81, 89]. Complementary to corners, blob-like[1]features are quite popular in the feature detection literature. Popular algorithms such as SIFT, SURF, Hessian-Laplace and Hessian-Affine produce responses at blob-like image structures [80]. Such detectors also share the very basic premise for feature detection, as they somehow rely on the entries of the Hessian matrix for feature detection:

$$\mathbf{H} = \begin{bmatrix} I_{xx}(x, y; \sigma_D) & I_{xy}(x, y; \sigma_D) \\ I_{xx}(x, y; \sigma_D) & I_{yy}(x, y; \sigma_D) \end{bmatrix} \tag{3.4}$$

where $I_{xx}(x, y; \sigma_D)$ is the second-order Gaussian smoothed derivative in the x direction. $I_{yy}(x, y; \sigma_D)$ and $I_{xy}(x, y; \sigma_D)$ are defined similarly.

The SURF feature detector relies on the determinant of the Hessian matrix for feature detection and scale selection [10, 11]. In SIFT, the Difference of Gaussians (DoG) function approximates the trace of the Hessian matrix, that is, the Laplacian of Gaussian (LoG) [80].

In the sequel, the SIFT detection process is briefly presented. The objective is to provide some more background on SIFT as it will be used in Chapter 4 to obtain a set of visual features.

[1]Smooth image regions which are brighter or darker than the background and stand out from their neighborhood [90, 91]

3.2.1 SIFT detector

A remarkable work in feature detection and description has been reported in [7, 8] which proposes a method to transform an image into a large collection of stable local features. Those features are invariant to image scaling, translation, rotation and robust to illumination and 3D viewpoint change.

The feature detection process in SIFT builds upon important developments and findings in the field of scale-space theory [88, 92]. The fact that objects in the world appear differently depending on the scale of observation and the need to cope with size variations resulting from projecting those objects in the image plane have led to the development of frameworks for describing image structures at different scales. In particular, the scale-space representation of an image is defined as a function $L(x, y, \sigma)$ constructed convolving the input image with Gaussian kernels of various scales σ:

$$L(x, y; \sigma) = g(x, y, \sigma) * I(x, y) \tag{3.5}$$

where $g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$.

A scale selection mechanism is proposed in [92] by introducing the concept of scale-normalized derivatives. The author pointed out that in the absence of other evidence, the scale level at which some combination of normalized derivatives of the scale-space function assumes a local maximum over scales can be treated as expressing a characteristic length of a corresponding structure. Using those normalized derivatives the author has shown that the response of the differential operators used for features extraction could be made invariant to scale changes. In particular, for a blob-like feature detector based on the trace of the Hessian matrix corresponding to the scale-space function $L(x, y, \sigma)$, such normalization leads to:

$$\begin{aligned}
\mathrm{Tr}(\mathbf{H}_{norm}) &= \sigma^2 \left[L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma) \right] \\
&= \sigma^2 \left\{ \frac{\partial^2}{\partial x^2} \left[g(x, y, \sigma) * I(x, y) \right] + \frac{\partial^2}{\partial y^2} \left[g(x, y, \sigma) * I(x, y) \right] \right\} \\
&= \sigma^2 \left\{ \frac{\partial^2 g(x, y, \sigma)}{\partial x^2} * I(x, y) + \frac{\partial^2 g(x, y, \sigma)}{\partial y^2} * I(x, y) \right\} \\
&= \left[\sigma^2 \nabla^2 g(x, y, \sigma) \right] * I(x, y)
\end{aligned} \tag{3.6}$$

The term within square brackets is the so-called Laplacian of Gaussian (LoG). In order to provide computation efficiency, it has been show in [7, 8], resorting to the diffusion equation, that the LoG function could be approximated by the Difference-of-Gaussians (DoG) function :

$$g(x, y, k\sigma) - g(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 g(x, y, \sigma) \tag{3.7}$$

The author has argued that the DoG functions having scales differing by a constant factor already take into account the normalization factor σ^2 required for scale invariance and the constant term $(k-1)$ does not influence extrema detection. Consequently, one may approximate Equation 3.6 by:

$$\begin{aligned} D(x, y, \sigma) &= [g(x, y, k\sigma) - g(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3.8}$$

The SIFT detector searches for extremum (maximum or minimum) points in the DoG images $D(x, y, \sigma)$ as given in Equation 3.8. A pyramid structure of DoG images is constructed to localize stable features in space and scale. Figure 3.2 depicts schematically the pyramid construction. The input image is convolved with Gaussian kernels of increasing scale values. These scale values are set in a particular fashion to produce smoothed images separated by a constant factor k. The smoothed images are grouped in octaves. The input image is smoothed until doubling the scale σ; once this happens, the smoothed image is down-sampled to reduce computation and a new octave is created. The number of scales per octave as well as a pre-smoothing step applied prior to octave construction has been determined after a set of experiments. Adjacent smoothed images within each octave are subtracted to produce the DoG images (as shown on the left side of Figure 3.2).

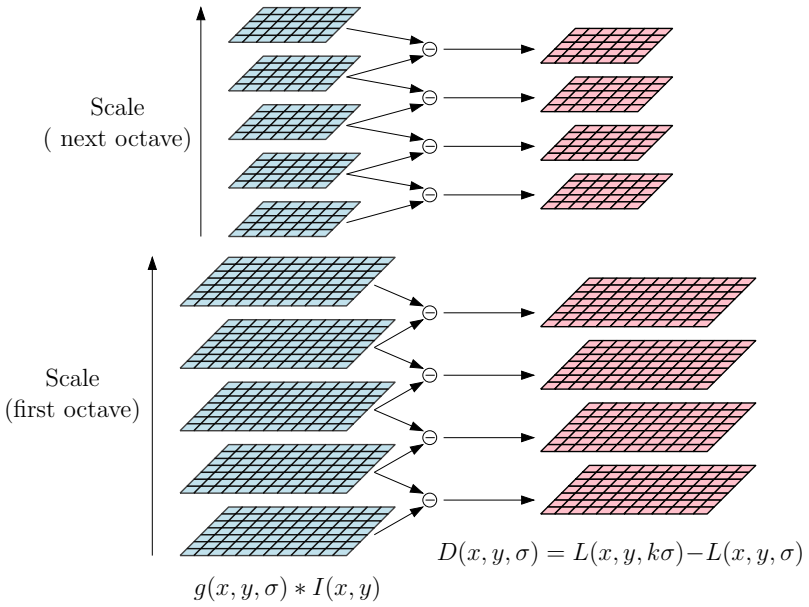


Figure 3.2: SIFT detection based on scale-space function[2].

[2]Based on a figure from [8].

Once the DoG image pyramid is constructed, keypoints are detected following the steps briefly described below:

- **Scale-space extrema detection:** in order to detect stable features both on space and scale, the amplitude of each sample point of a DoG image $D(x, y, \sigma)$ in the pyramid is compared with the one of its local neighboring samples. The comparison is made regarding the samples of the same DoG image as well as with those of adjacent DoG images as schematically shown in Figure 3.3. The red cross indicates the sample being tested and the green blocks its neighboring samples. If the sample amplitude at the 3-D coordinate (x, y, σ) is found to be smaller or larger than those of its neighborhood, this point (x, y, σ) is selected as a candidate keypoint location and is subjected to further analysis.

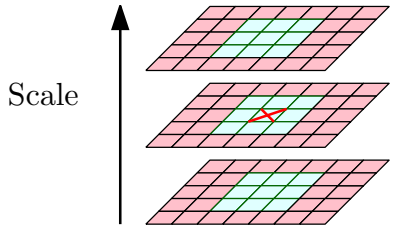


Figure 3.3: Sample amplitude comparisons for scale-space extrema detection.

- **Scale and location refinement:** the candidate 3-D coordinates (x, y, σ) detected in the previous step as local extrema are refined by fitting a quadratic function to the local sample points. To this end, $D(x, y, \sigma)$ is expanded using the Taylor series up to the quadratic terms. The Taylor expansion is shifted so that the origin is located at the sample point (x, y, σ). Taking the partial derivative with respect to $\mathbf{x} = (x, y, \sigma)$ and setting to zero leads to the extreme location $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = -\left[\mathbf{H}(D)\right]^{-1} \cdot \nabla^T(D) \tag{3.9}$$

  where the Hessian matrix $\mathbf{H}(D)$ and gradient vector $\nabla(D)$ of the DoG image at the 3-D coordinate (x, y, σ) are approximated using differences of neighboring sample points. Keypoints for which the absolute amplitude value $|D(\hat{\mathbf{x}})|$, evaluated at the extremum $\hat{\mathbf{x}}$, is less than a threshold, are discarded.

- **Edge response discard:** due to the strong response of $D(x, y, \sigma)$ along edges, the local curvature is checked to discard poor keypoint locations. A criterion based on the ratio of the principal curvatures is used for this purpose. The keypoint is kept only if:

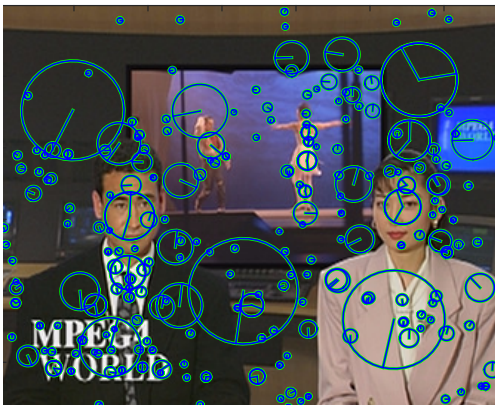
$$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{3.10}$$

where, the Hessian matrix $\mathbf{H}$ is computed at the detected scale and is given by:

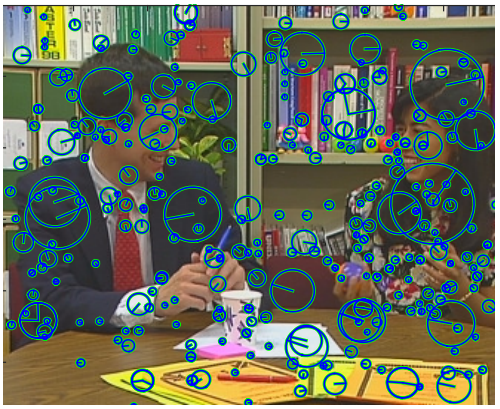
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{3.11}$$

- **Orientation assignment:** the final step is to assign one or more dominant orientations to each keypoint. To provide scale-invariance, this orientation assignment procedure uses the Gaussian smoothed image L closest to the detected scale. The objective is mainly to make the descriptor vectors rotationally invariant by representing them relative to those assigned dominant orientations. To this end, a histogram of gradient orientations is created for each keypoint by accumulating the gradient data of a local neighborhood. The histogram has 36 orientation bins, sampling the range of 360 degrees. Each sample added to the histogram is weighted by the gradient magnitude and by a weighting factor given by a Gaussian window. All orientation peaks within 80% of the highest peak produce a different keypoint differing by the orientation. A final fitting step is carried out in order to improve the peak orientation accuracy.

SIFT produces a set of keypoints following the computation steps briefly described above. Each one of those keypoints has a position (x, y), a scale (σ) and an angle (or orientation) (θ). To simplify the notation, each keypoint will be represented by $\mathbf{p} = \begin{bmatrix} x & y & \sigma & \theta \end{bmatrix}$. Figure 3.4 exhibits the set of detected keypoints superimposed on three example frames. The radii of the circles reflect the scale of the keypoints and the lines within the circles their orientations. One can notice that the number of detected keypoints and their spatial distribution somehow reflect the content within the frame.



(a) Sequence *News*, Frame 1.



(b) Sequence *Paris*, Frame 1.

(c) Sequence *Mobile*, Frame 1.

Figure 3.4: Keypoints detected with SIFT detector[3].

3.2.2 Feature detection assessment

As previously pointed out, the most desired property for local visual features is repeatability, that is, the ability to be repeatedly detected on two or more images depicting a common content, although differently acquired or processed. In this context, in order to assess the quality of the detected local features, the computation of a repeatability score has been proposed [80]. One starts from two images, let us say image A and image B, related by a homography H as schematically shown in Figure 3.5. A keypoint $\mathbf{p}_{A,i}$ in A and a keypoint $\mathbf{p}_{B,j}$ in B are said to correspond if the *overlap error* is less than a threshold ϵ, that is,

$$1 - \frac{\mathcal{A}(\mu_{A,i} \cap (H^T \mu_{B,j} H))}{\mathcal{A}(\mu_{A,i} \cup (H^T \mu_{B,j} H))} < \epsilon \tag{3.12}$$

where $\mu_{A,i}$ is a conic (ellipse or circle) defined as a function of the keypoint $\mathbf{p}_{A,i}$ in image A; $(H^T \mu_{B,j} H)$ is a conic $\mu_{B,j}$ defined as a function of keypoint $\mathbf{p}_{B,j}$ in image B and mapped on image A; and $\mathcal{A}(\cdot)$ the area. The areas of the intersection

[3]The Vlfeat [93] implementation has been used for detecting the superimposed keypoints.

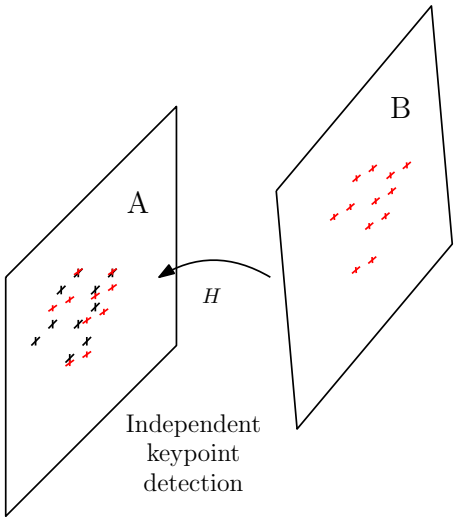


Figure 3.5: Independent keypoint detection on two images related by a homography.

$\mu_{A,i} \cap (H^T \mu_{B,j} H)$ and of the union $\mu_{A,i} \cup (H^T \mu_{B,j} H)$ are computed numerically. The threshold ϵ is commonly set to 0.4 [83].

The repeatability score r is computed as the ratio between the number of keypoint correspondences and the smallest number of detected keypoints in the two images. More precisely:

$$r = \frac{|R(\mathbf{p}_A, \mathbf{p}_B)|}{\min(|\mathbf{p}_A|, |\mathbf{p}_B|)} \tag{3.13}$$

where $|R(\mathbf{p}_A, \mathbf{p}_B)|$ is the number of keypoint correspondences fulfilling the criterion defined in Equation 3.12, $|\mathbf{p}_A|$ is the number of keypoints detected in image A, $|\mathbf{p}_B|$ is the number of keypoints detected in image B and $\min(\cdot, \cdot)$ returns the smallest of two numbers.

There are several papers in the literature for evaluating feature detectors, notably considering various test conditions and targeting applications as well as different feature detectors. The repeatability score is commonly assessed under image changes such as scaling, rotation, blur, illumination and compression. The image set used in those assessments usually consists of planar scenes or is acquired by cameras having the same center. This is so because one must provide a homography H relating the reference image and the transformed image.

Table 3.1 provides a qualitative summary of a few feature detectors, notably the type of underlying image structure detected, and the invariances to typical image changes. The strengths of the listed detectors regarding repeatability, localization accuracy, robustness and computational efficiency are also given. This table is a simplified version of the one provided in [80]. It is worthy noticing that the detectors aiming at fast computation such as SIFT and SURF provide a balance between

Table 3.1: Characteristics and strengths of a few feature detectors

Feature detector	Corner	Blob	Rotation invariant	Scale invariant	Affine invariant	Repeatability	Localization accuracy	Robustness	Efficiency
Harris	✓		✓			+++	+++	+++	++
Hessian		✓	✓			++	++	++	+
Harris-Laplace	✓	(✓)	✓	✓		+++	+++	++	+
Hessian-Laplace	(✓)	✓	✓	✓		+++	+++	+++	+
DoG (SIFT)	(✓)	✓	✓	✓		++	++	++	++
SURF	(✓)	✓	✓	✓		++	++	++	+++
Harris-Affine	✓	(✓)	✓	✓	✓	+++	+++	++	++
Hessian-Affine	(✓)	✓	✓	✓	✓	+++	+++	+++	++

invariance, repeatability and computational efficiency. Exhaustive assessments on feature detectors can be found in [81, 83, 94–96].

3.3 Local features description

The feature detection step outputs a set of repeatable image locations, detected in a rotation and scale invariant manner. Each one of those image locations has been assigned a location, a characteristic scale and a dominant orientation. The availability of distinctive and repeatable image locations which are likely to be detected on other images of the same scene or object, provides the very basic element to look for image region correspondences. In this context, the most common approach is to construct first a feature descriptor of the local image appearance based on some image property and to perform after descriptor vector matching.

A large number of local feature descriptors have been proposed in literature, naturally adopting different approaches for feature description, for instance, image derivatives and histogram-based descriptors [97, 98]. In particular, the description methods based on the histogram of the gradient data of an image area around a given keypoint present the best performance in matching tasks [8, 11, 98]. This approach, that first appeared in SIFT [7, 8], has been inspired in the response of neurons in the visual cortex. Since then, several other histogram-based description tools have been proposed in the literature such as PCA-SIFT [99], Gradient Location and Orientation Histogram (GLOH) [98], SURF [10, 11], Compressed Histogram of Gradient Orientation (CHoG) [23], only to name a few. As the ultimate goal is to produce correct descriptor matches, feature description tools are designed aiming at distinctive descriptor vectors at the same time having rotation-invariance, scale-invariance and robustness to local image deformations. In addition, one may also need to take into account the computational complexity; both the one for extracting the descriptor vector and the complexity incurred at the matching step as a result of adopting high-dimensional vectors.

Lately, binary descriptors have been a trend in local feature descriptors [86]. They target scenarios of low computational power and low memory consumption, as

well as simple and fast matching. Such description tools adopt as descriptor vectors binary strings resulting from simple intensity difference tests carried out on a sampling pattern around the keypoints. Remarkable works on binary feature descriptors include the Binary Robust Independent Elementary Features (BRIEF) [100], Binary Robust Invariant Scalable Keypoints (BRISK) [85] and Fast Retina Keypoint (FREAK) [101].

3.3.1 SIFT descriptor

This section gives a summary of the SIFT descriptor, mainly because it is still ranked among the best performing descriptors in matching tasks [98] and is the one adopted in Chapter 4 for feature detection and description.

The SIFT descriptor describes the visual features through an 128-D vector. These vectors capture the gradient information in a local square neighborhood around the keypoints, and are designed to be scale invariant, rotation invariant, and robust to illumination changes and positional shifts. The main aspects for obtaining the descriptor vector for each keypoint are briefly described in the sequel:

- **Gradient orientation and magnitude computation:** the gradient magnitude and orientation of the samples around the detected interest point are computed using the appropriate Gaussian smoothed image $L(x, y, \sigma)$ of the scale pyramid. Aiming rotation invariance, the gradient orientation is computed relative to the dominant orientation assigned to the keypoint (see Section 3.2.1). The gradient magnitude is weighted by a Gaussian function with the objective of assigning less weight to the samples far from the detected position. The left side of Figure 3.6 schematically shows the gradient data for the image region around the detected keypoint. The magnitude and orientation of the gradient for each sample are denoted by the length and orientation of the arrows.

- **Descriptor vector construction:** the square region around the keypoint is divided into 4×4 subregions (highlighted in red in Figure 3.6). For each subregion, a histogram with 8 bins is constructed, each bin corresponding to one of eight gradient orientations as schematically shown on the right of Figure 3.6. The gradient orientation of the samples in each subregion is quantized into one of those orientations and the weighted gradient magnitudes are accumulated. In order to cope with variations in sample location and gradient orientation, each gradient magnitude is distributed into adjacent histogram bins by performing trilinear interpolation [8, 35]. The descriptor vector is constructed by concatenating these 4×4 histograms, resulting in an 128-D vector for each detected keypoint.

- **Normalization and large gradient thresholding:** in order to reduce the effect of affine illumination changes on the descriptor vector, the constructed vector is normalized to unit length. After that, large descriptor vector components are clipped to a certain threshold with the objective to reduce the influence of large gradients resulting from non-linear illumination changes.

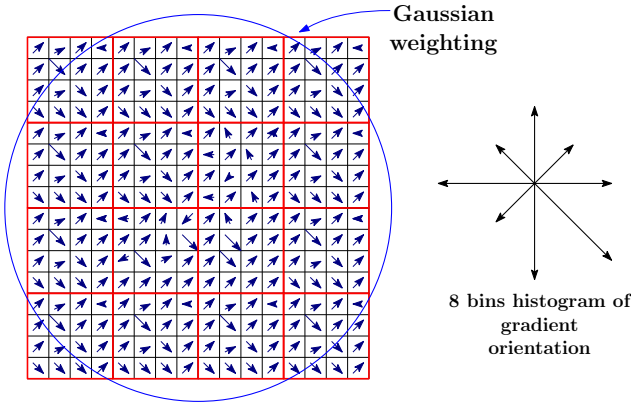
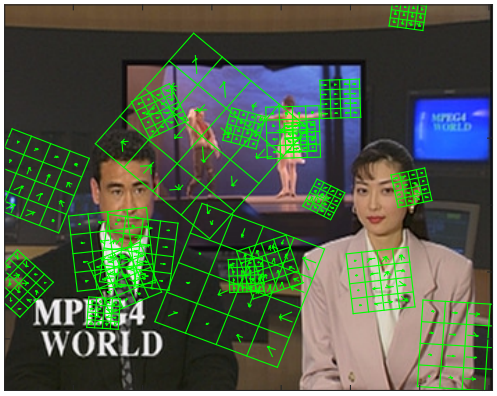
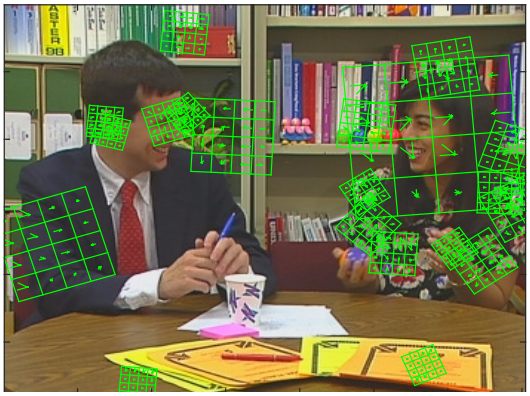


Figure 3.6: SIFT descriptor extraction.

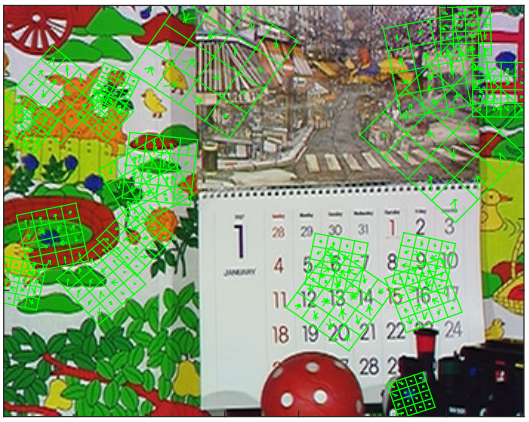
Figure 3.7 shows a few local image regions from which descriptor vectors are extracted for three example frames. A graphical representation of the computed histogram is superimposed where the sizes of the square regions denote the scale of the keypoint. At the end of the detection and description steps, SIFT outputs a set of repeatable and distinctive visual descriptors. Each feature has a position (x, y), a scale σ, an orientation θ and a descriptor vector $\mathbf{d} \in \mathbb{R}^{128}$ consisting of a histogram of gradient orientations.



(a) Sequence *News*, Frame 1.



(b) Sequence *Paris*, Frame 1.



(c) Sequence *Mobile*, Frame 1.

Figure 3.7: SIFT descriptor extraction from squared regions around the keypoints[4]. Only 20 keypoints are shown for each image.

3.3.2 Pairwise descriptor matching

Once one has the means to detect keypoints within an image and to describe the local appearance of each one of those keypoints by a descriptor vector, the groundwork for establishing correspondences between two images depicting a common scene or object has been set. Let us consider a working example, with $\mathcal{D}_A$ being the set of local features detected and extracted from an image A, each one composed of a keypoint $\mathbf{p}_{A,i}$ and the associated descriptor vector $\mathbf{d}_{A,i}$. Similarly, let $\mathcal{D}_B$ be the set of features for an image B, each one composed of $\mathbf{p}_{B,i}$ and $\mathbf{d}_{B,j}$. The objective is to establish correspondences between the two images by matching their descriptors in a pairwise manner. A descriptor $\mathbf{d}_{B,j}$ from image B is deemed to match the descriptor $\mathbf{d}_{A,i}$ from image A if it minimizes a distance measure [8, 98, 102]. Several distance measures have been used for matching descriptor vectors. Besides simplicity, the adopted distance measure depends on the sort of information captured in the de-

[4]The Vlfeat [93] implementation has been used for detection and extraction.

scriptor vector. Differential-invariant and moment-invariant based descriptors, for instance, are matched using the Mahalanobis distance as dissimilarity measure [98], whereas binary descriptors use Hamming distance [85, 100]. As for histogram-based descriptors, common dissimilarity measures include L^2-norm [8], χ^2 distance and Earth Mover's Distance (EMD) [102]. From a broad view, distance measures for descriptors matching can be categorized into bin-to-bin distances and cross-bin distances [102, 103]. For SIFT descriptors [7, 8], because of its simplicity, it is common the use of the L^2-norm of the error. Therefore, a simple matching criterion consists of minimizing this norm as follows:

$$j^* = \arg\min_j \parallel \mathbf{d}_{A,i} - \mathbf{d}_{B,j} \parallel_2 \tag{3.14}$$

Such simple matching criterion is likely to produce many false descriptor matches [8]. A simple measure has been proposed for SIFT aiming to discard those false matches [8, 104]. It consists of comparing the distance of the closest descriptor in the image B, denoted by $\mathbf{d}_{B,j*}$ and the distance of the second-closest one, denoted $\mathbf{d}_{B,j'}$ as schematically show in Figure 3.8.

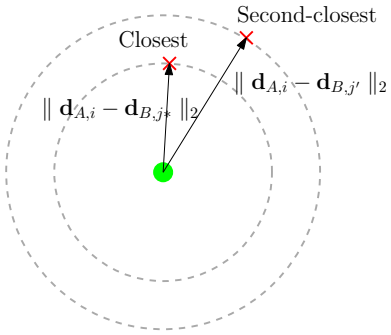


Figure 3.8: Ratio test criterion for discarding false matches.

The matches for which the ratio between the distance of the closest descriptor and the distance of the second-closest descriptor is greater than a 0.8 are discarded. More precisely:

$$\frac{\parallel \mathbf{d}_{A,i} - \mathbf{d}_{B,j^*} \parallel_2}{\parallel \mathbf{d}_{A,i} - \mathbf{d}_{B,j'} \parallel_2} > 0.8 \tag{3.15}$$

The threshold 0.8 has been determined experimentally in the context of an object recognition task [8]. In order to further reduce the number of false matches, one may also perform pairwise descriptor matching from image B to image A and assume as correct matches those descriptor matches that occur in both directions (cross-matching or 2-way matching) [105]. Figure 3.9 illustrates schematically the cross-matching criterion for the 2-D matching case.

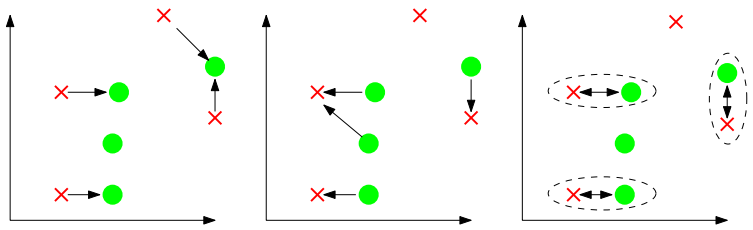
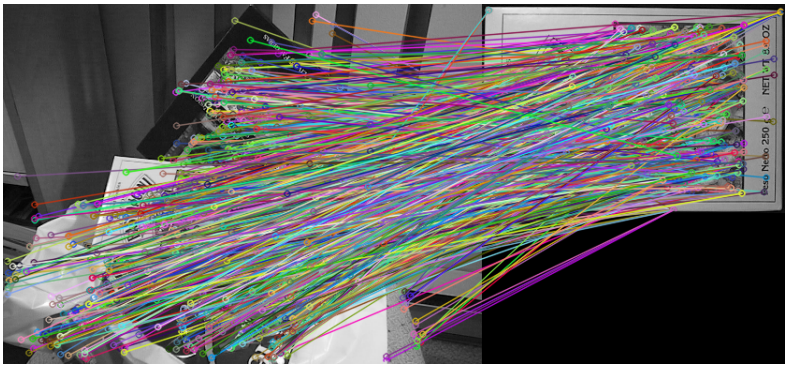


Figure 3.9: 2-D cross matching criterion.

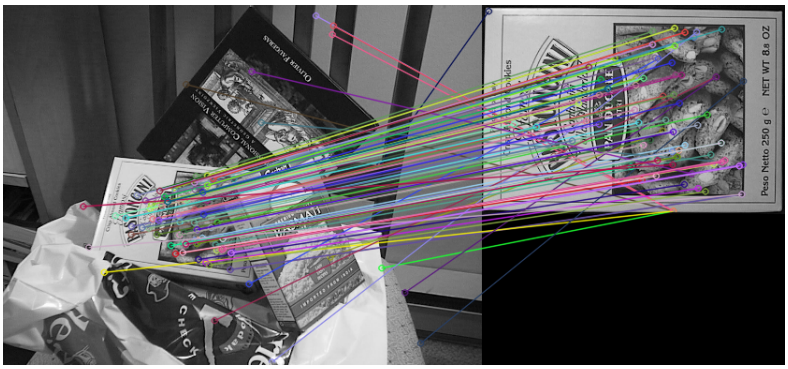
In some more geometrically constrained applications such as matching images of planar objects, in addition to the ratio test above, a geometric consistency check may be carried out [11, 14, 22, 106]. To this end, a homography is estimated by means of robust estimation methods such as RANSAC. The pairwise matches that do not agree with the homography are discarded.

Figure 3.10 shows the pairwise matches resulting from adopting the matching criteria described above.

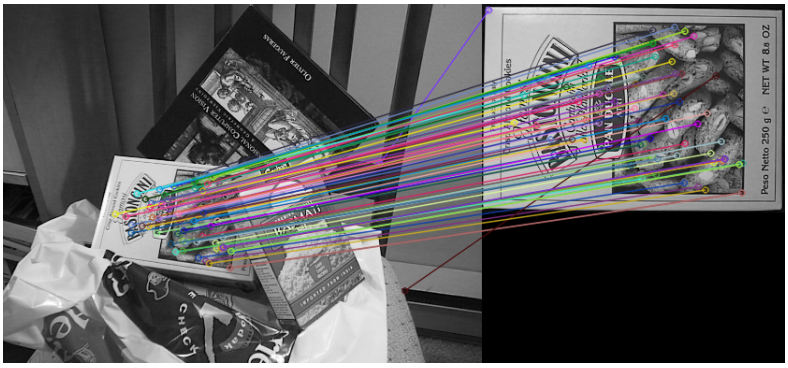
(a) Original image pair.



(b) All pairwise matches obtained by using L^2-norm as distance measure.



(c) Pairwise matches after applying the ratio criterion.



(d) Pairwise matches after applying the cross-matching criterion.

Figure 3.10: Pairwise matches resulting from adopting different matching criteria.

3.4 Local visual feature coding

The problem of compacting visual features extracted from images has been tackled by researchers in several ways: through dimensionality reduction [99], specially designed compressed feature descriptors such as the Compressed Histogram of Gradient (CHoG) [23], transform coding [21, 22] and binary descriptors [24, 101]. Naturally, the result should preserve the desirable properties of the descriptors, and yet being computationally easy to obtain. More recently, the attention has turned to coding visual features extracted from video sequences. In [14], the authors have proposed a feature coding framework with various coding modes, including intra-frame and inter-frame, with and without decorrelating transforms.

This section presents a brief review of coding schemes for visual features extracted from video sequences as well as the rate-distortion performance of various coding setups. This description is mostly based on the works reported in [14, 21, 22]. A detailed treatment can be found in [107]. To settle any doubt, it is worth reiterating that a visual feature includes the keypoint and the descriptor vector as reviewed in Sections 3.2 and 3.3, respectively. The coding schemes described can be classified as intra-frame coding schemes and inter-frame coding schemes. Usually, the best coding scheme from the Rate-Distortion (RD) optimization point of view is typically one appropriately combining the intra- and inter-frame coding schemes. In intra-frame coding schemes, the set of visual features $\mathcal{D}_n$ extracted from the n-th frame of a video sequence is coded independently from those of other frames. On the other hand, in inter-frame coding schemes, the redundancy between descriptors of neighboring frames can be exploited to save bit rate. These schemes are inspired in the predictive coding tools employed in traditional video encoders.

3.4.1 Intra-frame coding schemes

In intra-frame coding schemes, the set of features $\mathcal{D}_n$ of each frame is coded independently. Nevertheless, the correlation between the various components within a descriptor vector may be exploited. Each feature has two components, namely the descriptor vector $\mathbf{d}_{n,i}$ describing the image patch centered at the detected keypoint and the keypoint itself $\mathbf{p}_{n,i} = \begin{bmatrix} x & y & \sigma & \theta \end{bmatrix}$ consisting of location (x, y), scale σ and the dominant orientation θ. Both the keypoint $\mathbf{p}_{n,i}$ and the descriptor vector $\mathbf{d}_{n,i}$ should be coded. Each element of the keypoint is quantized with a quarter unit precision and entropy coded. The descriptor vector part $\mathbf{d}_{n,i} \in \mathcal{D}_n$ of each feature is scalar quantized and entropy coded after an orthonormal transformation. Figure 3.11 illustrates the general idea of intra-frame coding schemes. Each step is briefly described in the sequel:

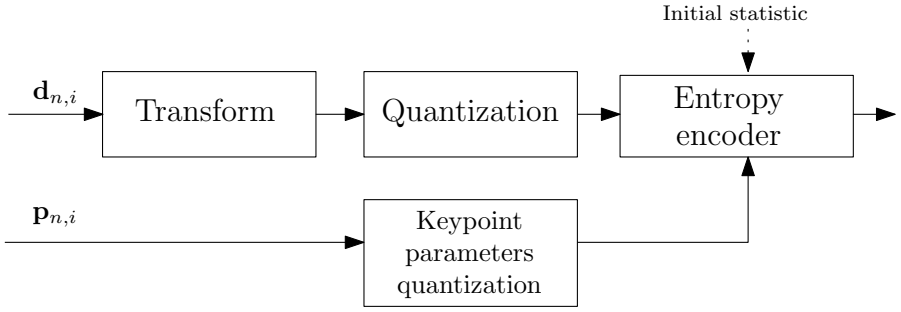


Figure 3.11: Intra coding scheme.

- **Transform:** the simplest coding approach is simply to quantize and entropy code the descriptor vector, that is, skip the transform step. An alternative option is to use the Karhunen-Loève (KL) transform, which is known to achieve maximal energy compaction, suitable for compression; and has been successfully employed in descriptor coding as reported in [14, 21, 22]. A collection of descriptors extracted from training video sequences is used to estimate the covariance matrix $\Sigma_{\mathbf{d}}$ in order to calculate the KL transform. Since the descriptor vector used has dimensionality 128, $\mathbf{KL}_{intra}$ is an 128×128 matrix. After the transform, one generates $\mathbf{c}_{n,i}^{\text{INTRA}} = \mathbf{T}\mathbf{d}_{n,i}$, where $\mathbf{T} \in \{\mathbf{I}, \mathbf{KL}_{intra}\}$ (the identity $\mathbf{I}$ is only to simplify the notation). Notice that the transform step is not mandatory, the descriptor vector may go directly through to scalar quantization and entropy coding. That is, when $\mathbf{T} = \mathbf{I}$ implies that $\mathbf{c}_{n,i}^{\text{INTRA}} = \mathbf{d}_{n,i}$

- **Quantization:** a straightforward scalar quantization is used [14, 21, 22]. Each transform coefficient (or each descriptor component, in case of not applying the transformation) is quantized as:

$$\tilde{c}_{n,i,j} = \text{round}\left(\frac{c_{n,i,j}}{QS}\right) QS \tag{3.16}$$

  where $c_{n,i,j}$ is j-th component of $\mathbf{c}_{n,i}^{\text{INTRA}}$ (the vector of transform coefficients) and QS is the quantization step size. The function round(x) rounds x to the nearest integer.

- **Entropy coding:** an arithmetic coding is employed in order to entropy code the quantized transform coefficients resulting from applying the transform and quantization steps to the descriptor vector, as well as the keypoint parameters position, scale and orientation. An initial statistical model is set up to the quantized transform coefficients, and during the coding process the statistical model is adaptively updated. Training sequences are coded to generate the statistics for each transform coefficient. This is also the case when the transform step is skipped. The keypoint location, scale and orientation are coded

using a uniform probability model.

In summary, as described briefly above, the descriptor vector can be coded in two ways, namely *Intra* and *Intra-KLT*. The transform applied to the descriptor vector is the main difference between the two coding modes.

3.4.2 Inter-frame coding schemes

The inter-frame coding schemes are inspired in traditional video encoders such as H.264/AVC and HEVC. A predictive scheme is used to take advantage of the repeatability property of robust local image features as those detected by SIFT [8] and SURF [11] in addition to the smooth change of the captured scene. The set of descriptors coded from last frame $\hat{\mathcal{D}}_{n-1}$ can be used as a prediction to current frame descriptors $\mathcal{D}_n$. First, a matching for each descriptor vector $\mathbf{d}_{n,i}$ is found for the encoder to take the prediction residue between the current descriptor $\mathbf{d}_{n,i}$ and the matching descriptor vector $\mathbf{d}_{n-1,k^*}$. The prediction residue is transformed, followed by quantization and entropy coding. A predictive scheme is also adopted to code the position, scale and orientation. Figure 3.12 shows a block diagram of the inter-frame coding schemes. Each step is described in the sequel:

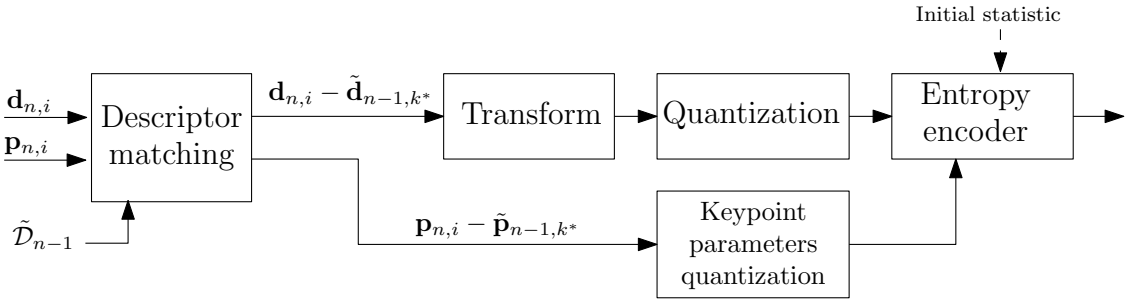


Figure 3.12: Inter-frames coding.

- **Descriptor matching:** the encoder performs a search for a matching descriptor vector decoded from the reference set $\tilde{\mathcal{D}}_{n-1}$. The nearest descriptor $\tilde{\mathbf{d}}_{n-1,k*}$ is found using the distance metric:

$$\tilde{\mathbf{d}}_{n-1,k*} = \arg\min_{\tilde{\mathbf{d}}_{n-1,k}} \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k}\|_2 \tag{3.17}$$

$$\text{subject to } \begin{cases} x_{n,i} - \tilde{x}_{n-1,k} \leq w_x; y_{n,i} - \tilde{y}_{n-1,k} \leq w_y \\ \sigma_{n,i} - \tilde{\sigma}_{n-1,k*} \leq w_s \end{cases}$$

where $\tilde{\mathbf{d}}_{n-1,k} \in \tilde{\mathcal{D}}_{n-1}$, P is the vector dimensionality, whereas w_x, w_y and w_s are the search windows and $\|\cdot\|_2$ refers to the L^2-norm.

Interest points in a scene have high probability to be detected repeatedly in a frame sequence with smooth changes in position. Therefore, the matching step is constrained to reduce computational complexity. The matching search is usually performed within a spatial window of $w_x = w_y = 30$ pixels in the horizontal and vertical directions and scale window of $w_s = 5$.

With regard to location, scale and orientation, only the prediction errors and the prediction reference are coded. In other words, the differences of position $(x_{n,i} - \tilde{x}_{n-1,k^*}; y_{n,i} - \tilde{y}_{n-1,k*})$, scale $\sigma_{n,i} - \tilde{\sigma}_{n-1,k*}$ and orientation $\theta_{n,i} - \tilde{\theta}_{n-1,k*}$ are quantized and entropy coded. The coding order of the feature descriptors $\mathcal{D}_n$ of the current frame is set with regard to the coding order of the matching reference descriptors $\tilde{\mathcal{D}}_{n-1}$, and a differential scheme is used to code the prediction reference, more details can be found [14].

- **Transform:** the simplest approach for coding the descriptor vector residue $\mathbf{r}_{n,i} = \mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*}$ is simply to quantize and entropy code the descriptor vector residue, that is, skip the transform step. Alternatively, the encoder may apply a KL transform before the quantization and entropy coding steps. The procedure to obtain the $\mathbf{KL}_{inter}$ is similar to that used to obtain $\mathbf{KL}_{intra}$ as described above. However, is this case, a set of prediction residues should be collected in order to obtain the covariance matrix. Only prediction residues which satisfy $\|\mathbf{d}_{n,i} - \mathbf{d}_{n-1,k*}\|_2 < \|\mathbf{d}_{n,i}\|_2$ are collected for the purpose of obtaining the transform. This procedure is done using a training sequence. The vector of transform coefficients resulting from applying the transform to the descriptor vector residue $\mathbf{c}_{n,i}^{\text{INTER}} = \mathbf{T}(\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*})$ is then scalar quantized and entropy coded.

- **Descriptor residue quantization:** similar to the intra-frame coding schemes, a scalar quantizer is used for inter-frame schemes [14]. However, in this case the transformed descriptor vector residue $\mathbf{c}_{n,i}^{\text{INTER}} = \mathbf{T}(\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n-1,k^*})$ is quantized, where as described before $\mathbf{T} \in \{\mathbf{I}, \mathbf{KL}_{inter}\}$. The quantization is performed as defined in Equation 3.18.

$$\tilde{c}_{n,i,j} = \text{round}\left(\frac{c_{n,i,j}}{QS}\right) QS \tag{3.18}$$

where $c_{n,i,j}$ is j-th component of $\mathbf{c}_{n,i}^{\text{INTER}}$ and QS is the quantization step size.

The position, scale and orientation prediction errors are quantized and entropy coded.

- **Entropy coding:** arithmetic coding is used to entropy code the quantized transform coefficients resulting from applying the transform to the descriptor vector prediction residue as well as for coding the position, scale and orientation prediction errors. A training step is conducted in order to collect an initial statistical model for the transform coefficients. An initial probability is assigned to the transform coefficients as well as to the position, scale and orientation prediction errors. The encoder can update the probability during execution. This is also the case when the transform step is skipped.

In summary, as briefly described above, the descriptor vector residue can be coded in two ways, namely *Inter* mode and the *Inter-KLT* mode, either using or not a KL transform step.

3.4.3 Rate-distortion optimization

While Sections 3.4.1 and 3.4.2 have described the intra-frame and inter-frame coding schemes, respectively, the best coding solution is obtained by appropriately combining these coding schemes to better exploit the specific correlations associated to each descriptor vector.

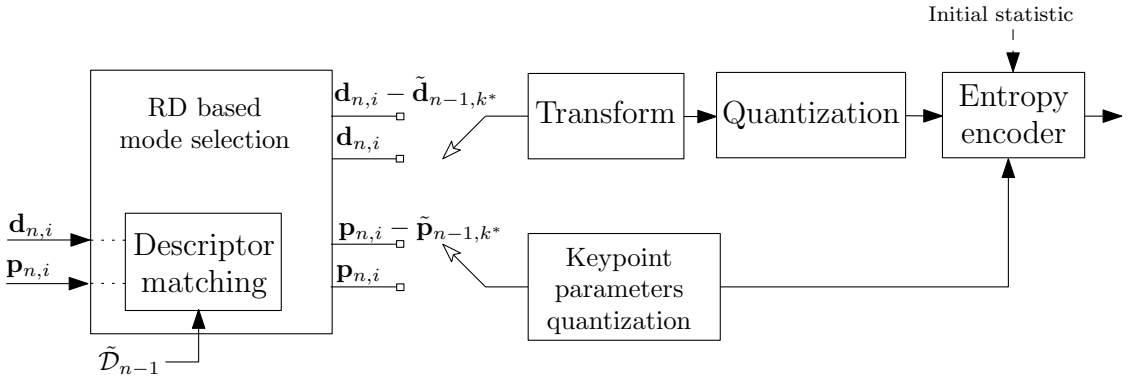


Figure 3.13: Rate-distortion optimization based encoder.

The encoder performs an RD optimization aiming to reach high fidelity with the smallest possible rate cost. Among the enabled coding modes the encoder chooses the coding mode which gives the minimum Lagrangian cost (see Figure 3.13). The cost function for intra coding mode is defined as:

$$J_{\text{INTRA}} = \frac{1}{\sqrt{P}} \|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2 + \lambda (R^{\mathbf{p}^{\text{INTRA}}_{n,i}} + R^{\mathbf{d}^{\text{INTRA}}_{n,i}}) \tag{3.19}$$

where $R^{\mathbf{p}^{\text{INTRA}}_{n,i}}$ is the cost to code the keypoint parameters position, scale and orientation, and $R^{\mathbf{d}^{\text{INTRA}}_{n,i}}$ is the cost to code the description vector. Note that in case

of using the KL transform, the rate $R^{\mathbf{d}^{\text{INTRA}}_{n,i}}$ is the rate spent to code the vector of transform coefficients. The cost function for the inter coding mode is defined as:

$$J_{\text{INTER}} = \frac{1}{\sqrt{P}}\|\mathbf{d}_{n,i} - \tilde{\mathbf{d}}_{n,i}\|_2 + \lambda(R^{\mathbf{p}^{\text{INTER}}_{n,i}} + R^{\mathbf{d}^{\text{INTER}}_{n,i}}) \tag{3.20}$$

where $R^{\mathbf{p}^{\text{INTER}}_{n,i}}$ is the cost to code the position, scale, orientation prediction errors as well as the prediction reference, and $R^{\mathbf{d}^{\text{INTER}}_{n,i}}$ is the cost to code the descriptor vector with respect to the matched reference descriptor vector. Again, the rate to code the descriptor depends on which transform was chosen.

The Lagrange multiplier λ controls the rate-distortion trade-off. Experiments were conducted in [14], inspired by [2], to obtain the optimal λ value. A rule of thumb adopted was $\lambda(QS) = 1.8 \cdot 10^{-4} QS^2 + 0.1$, where QS is the quantization step size.

Besides the rate necessary to code the descriptor vector and associated information such as position, scale and orientation, it is necessary to code the selected coding mode in the rate-distortion optimization. Moreover, to know the number of descriptors used for each frame, the encoder also needs to send a end-of-frame flag.

3.4.4 Results and discussion

The performance of coding schemes for descriptor vectors should be evaluated taking into consideration how much an encoded descriptor vector is effective in typical matching tasks. In this sense, RD results tend to have little meaning. In spite of this, it has been reported in the literature that there is a strong correlation between a descriptor's performance in typical matching tasks and its RD results. In fact, it was pointed out in [21] that at 15 dB of SNR the descriptor's rate-accuracy performance saturates. Similarly, in [14] it is also stated that the matching score saturation is achieved at 15 dB of SNR. Moreover, it was shown in [22] that MSE is good a predictor for both image and descriptor matching error, and that the SURF and SIFT descriptors achieve near-perfect image matching and retrieval below 2 bits per descriptor component. Therefore, the coding schemes are evaluated below from the RD perspective.

The previous sections have briefly described a framework to code descriptors and keypoints extracted from a video sequence. The framework includes intra- and inter-frame coding schemes, and in the RD optimization mode the encoder can decide the best coding strategy for each descriptor. A complete description and rate-distortion evaluation of those coding schemes can be found in [14, 107].

The rate-distortion performance of a particular implementation of the framework is presented below. The following coding setups were tested:

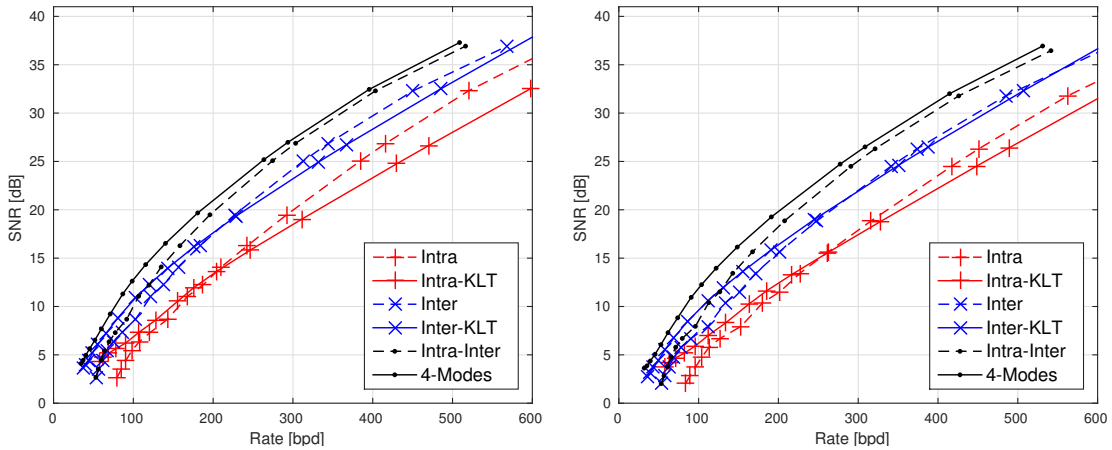
- *Intra*: all descriptor vectors are coded with intra coding mode, $\mathbf{T} = \mathbf{I}$.
- *Intra-KLT*: all descriptor vectors are coded with intra coding mode, $\mathbf{T} = \mathbf{KL}_{intra}$.
- *Inter*: the descriptor vectors are coded with inter coding mode, $\mathbf{T} = \mathbf{I}$. Exceptions are the descriptor vectors for which the matching step was not able to find any reference in the search window, including those descriptor vectors of the first frame. In this case, the descriptor vectors are coded using *Intra* mode.
- *Inter-KLT*: same as the *Inter* coding mode above but $\mathbf{T} = \mathbf{KL}_{inter}$.
- *Intra-Inter*: the encoder performs rate-distortion optimization with the *Intra* and *Inter* modes and chooses the mode with lowest cost.
- *4-modes*: the encoder performs rate-distortion optimization with the *Intra*, *Intra-KLT*, *Inter* and *Inter-KLT* modes and chooses the mode with lowest Lagrangian cost.

Figures 3.14 and 3.15 show the encoder performance for SIFT and SURF descriptors, respectively. As expected, when all coding modes are available, the encoder can choose the best coding strategy for each descriptor resulting in better overall performance.

For SIFT descriptors, the *Intra-KLT* mode achieves higher coding efficiency than *Intra* only in low bit rates. Similar behavior is observed when comparing inter-frame encoding modes, *Inter-KLT* outperforms *Inter* only in low bit rates. This corroborates the results reported in [22], and is a consequence of the non-Gaussianity of individual descriptor components.

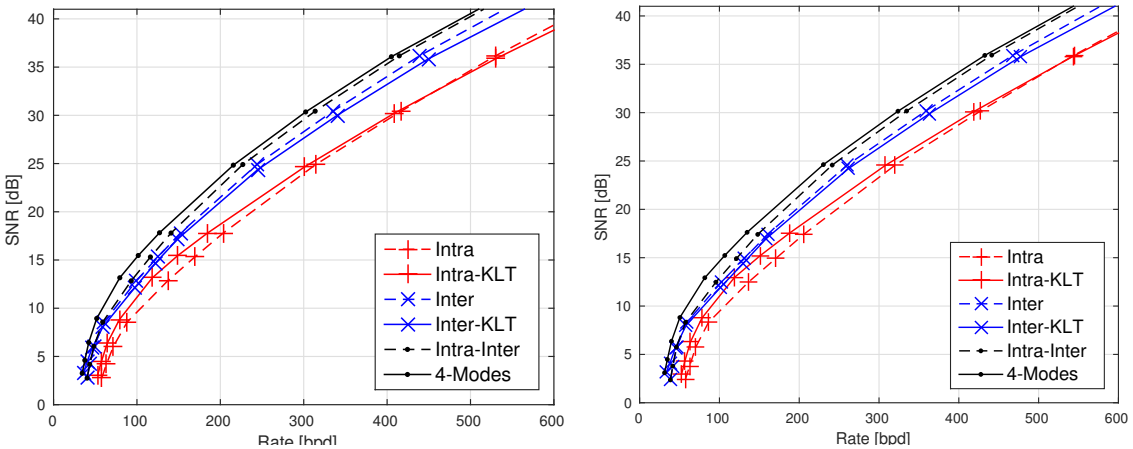
In case of SURF descriptors, the *Intra-KLT* mode outperforms the *Intra* coding mode in almost all bit rates. On the other hand, the performance of *Inter-KLT* is worse than the one of *Inter*, that is, applying KL transform to descriptor residues is detrimental to coding performance.

Using adaptively the various coding modes give better results than intra or inter schemes individually for both feature descriptors.



(a) Sequence *Foreman*. (b) Sequence *Mobile*.

Figure 3.14: Comparative performance for SIFT descriptor coding.



(a) Sequence *Foreman*. (b) Sequence *Mobile*.

Figure 3.15: Comparative performance for SURF descriptor coding.

3.5 Final remarks

This chapter has reviewed the detection and descriptor extraction processes followed for characterizing visual content in terms of local visual features. In particular, it has examined in more detail the SIFT detection and descriptor extraction steps. A coding framework has been described for coding the descriptors and the associated keypoints extracted from video sequences. In this coding framework, an RD optimized coding scheme adopting both intra- and inter-coding modes has been shown to perform the best according to the experiments.

The objective of this chapter was to provide the groundwork for the dual-purpose video coding solution proposed in Chapter 4, in which, an unified pixel-based and feature-based video coding solution is proposed targeting scenarios where visualization and searching needs are required to be addressed. This is motivated by the

current role of digital video signals in enabling more powerful applications.

Chapter 4

Dual-purpose video coding framework

This chapter presents the proposed Dual-Purpose Video Coding (DPVC) solution, and starts by briefly restating the context and motivations discussed in Chapter 1 underpinning this work. After that, the architecture and walkthrough of the proposed video coding solution are presented in Section 4.2, whereas Section 4.3 presents the most novel and technically original coding modules. The performance assessment of the proposed DPVC is carried out in Chapter 5 under meaningful test conditions, moreover, DPVC is compared to the state-of-the-art HEVC standard.

4.1 Introduction

As discussed in Chapter 1, applications that consider visualization and searching needs are becoming very popular together. In this context, the HATC approach has recently attracted attention because it attempts to overcome the limitations of the CTA and ATC [14–16]. In the former, despite allowing visualization, the compression process has a detrimental effect on the extracted visual features, which in turn impairs the visual analysis performance. The latter limits the range of applications by not enabling visualization. In previous HATC works, pixel and feature-based representations are essentially designed and used independently from each other, meaning that the feature-level data (targeting searching) is not exploited to aid the pixel-level coding (targeting visualization) and vice-versa. But this scenario is starting to change. In [20], a hybrid framework for jointly coding the feature descriptors and visual content is proposed, exploiting their interaction. While the feature descriptors are efficiently represented by taking advantage of the structure and motion information in the compressed video stream, the already compressed descriptors can be used to further improve the video compression efficiency by applying feature

matching based affine motion compensation. The novel video coding solution proposed in this chapter also adopts the HATC approach; however, differently from [20], the proposed solution explicitly codes just the keypoint data detected at the encoder as the descriptors themselves are extracted at the decoder using the keypoint-based reconstructed f-frames.

The proposed DPVC employs a hybrid approach where pixel-based and feature-based coding are combined to provide efficient video coding solution targeting both visualization and searching needs. The pixel-based processing is built upon the state-of-the-art HEVC, reviewed in Chapter 2, to code the so-called k-frames and to code the residue for the f-frames in the enhancement layer. As for the feature-based processing used for coding the f-frames in the base layer, the SIFT features, discussed in Chapter 3, are used for characterization in terms of local features. The coding scheme to code keypoint matches selected as being beneficial in the joint optimization routine is inspired in the keypoint coding part of the feature coding framework described in Section 3.4. The descriptors are extracted at the decoder using the keypoint-based reconstructed f-frames as they are less harmed by coding artifacts [15, 31]. The proposed solution is based on a flexible joint Lagrangian optimization framework where pixel-based and feature-based processing are combined to find the most appropriate trade-off between the visualization and searching performances. Moreover, the proposed solution provides quality scalability for the f-frames and some degree of compatibility with the latest video coding standard HEVC with the k-frames.

4.2 Architecture and walkthrough

This section presents the architecture and walkthrough of the proposed Dual-Purpose Video Coding (DPVC) solution, which combines pixel-based and feature-based coding to provide a powerful and efficient coding framework towards both visualization and searching. While the pixel-based component provides backward compatibility with the most efficient visualization-targeted video coding standard, the feature-based component boosts the searching performance by providing precise keypoint locations, extracted from the original, uncompressed video content. By targeting simultaneously two key functionalities, the dual-purpose coding process has to consider both a visual quality distortion, D_V, and a descriptor matching distortion, D_M, which assess the visualization and searching performances, respectively. The proposed dual-purpose coding architecture is presented in Figure 4.1 and its processing walkthrough is explained in the sequel:

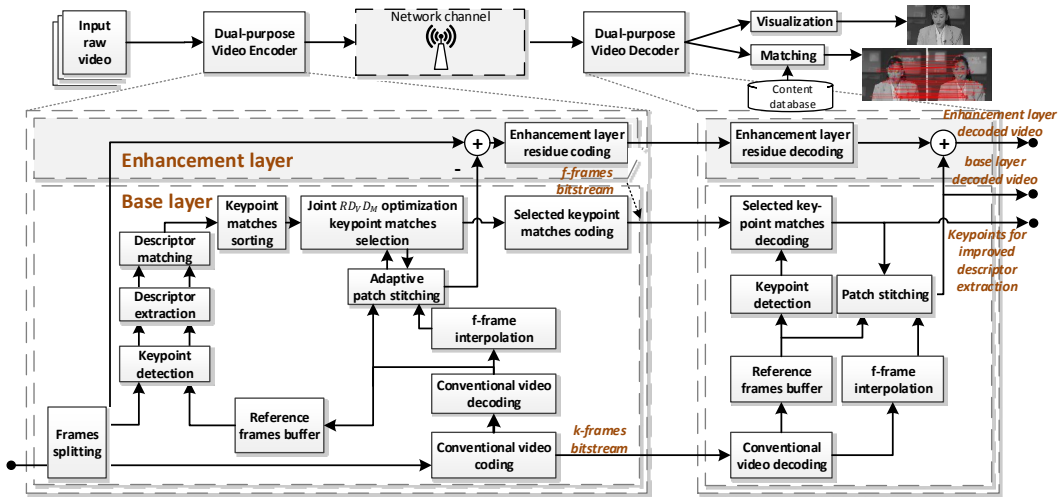


Figure 4.1: Block diagram of the proposed Dual-Purpose Video Coding (DPVC) solution.

4.2.1 Encoder

- **Frame splitting**: the original video frames are split in two sets, namely the so-called k-frames and f-frames. While the k-frames are coded using a conventional video coding solution, thus providing some backward compatibility, the f-frames are coded using a feature-based approach, explicitly making the overall coding framework searching friendly. The frames are arranged in a Group Of Pictures (GOP) structure where a k-frame is periodically inserted among the f-frames. A GOP includes a k-frame and the set of f-frames preceding the next k-frame; for GOP of size 2, the k-frames and f-frames alternate.

- **(k-frames) Conventional video coding and decoding**: after frame splitting, the k-frames are Intra coded and decoded using a standard video codec; in this case, the state-of-the-art HEVC standard is used [3]. The k-frames may also be coded using some conventional Inter coding solution, e.g. some HEVC profile, eventually providing a different trade-off between compression efficiency, random access, error resilience and coding complexity. The decoded k-frames play a central role in the proposed dual-purpose coding solution since they not only provide some backward compatibility but they also provide the references for the efficient coding of the f-frames.

- **f-frame interpolation**: to restore the original frame rate and effectively estimate regions with smoother spatial and temporal evolutions, an initial estimate of the f-frames is obtained by interpolating them from the neighboring decoded reference frames, typically k-frames. This is performed both at encoder and decoder using a block-based motion compensated frame in-

terpolation algorithm using as references the closest past and future available decoded reference frames. To this end, the algorithm proposed in [108] has been adopted. For GOP sizes longer than 2 (typically powers of 2 sizes), a hierarchical interpolation structure is used, making previously decoded f-frames the reference frames for other f-frames. In the simplest case, that is, GOP of size 2, only k-frames can play the role of references frames. In the following, for simplicity, the explanation is restricted to the more intuitive GOP of size 2 case but the extension to longer GOP sizes is straightforward.

- **Reference frames buffer**: the reference frames buffer includes the so-called reference frames, which provide texture patches to improve the initially estimated f-frames. The decoded k-frames are naturally the most common reference frames (usually one or more past k-frames and eventually a single future k-frame to limit the delay).

- **Keypoint detection and descriptor extraction**: to determine the best patches from the reference frames to improve the interpolated f-frames, the original f-frames and the decoded reference frames feed a keypoint detection module which identifies the most distinctive positions inside a frame in terms of feature-based characterization, and thus searching performance. For each keypoint, a descriptor is extracted to capture the local image patch. The objective is that only a parsimoniously selected number of keypoint matches are conveyed to the decoder as there is an associated rate cost. Their aim is twofold: First, they indicate the areas in the approximation of the f-frames estimated by interpolation that may have its quality more improved for visualization using decoder-available patches. Second, they indicate distinctive f-frame areas likely to be correctly matched to a visual content database available at the receiver side as this improves the searching performance. In this work, SIFT [7, 8] is used for keypoint detection and descriptor extraction.

- **Descriptor matching**: to be able to improve the f-frames estimates with patches from the reference frames, the f-frames descriptors are matched to the reference frames descriptors using the Euclidean distance as matching metric. The intuition is that the matching descriptor pairs represent regions with similar visual content in the f-frames and reference frames.

- **Adaptive patch stitching**: to increase the overall f-frames quality, some interpolated f-frame regions may be improved with appropriate matching patches from the reference frames by performing patch stitching using the Poisson stitching technique proposed in [109]. To determine the best visual quality improvement impact associated to each keypoint match, a scale parameter

factor may be used, and thus its best value has to be adaptively selected depending on the specific content. This selection is made by assessing the Mean Squared Error (MSE) reduction considering the image patch centered at a certain reference frame keypoint location seamlessly stitched to the corresponding matching f-frame keypoint location and the corresponding original f-frame region. Before patch stitching, the reference frame patch is conveniently rotated, scaled and translated to better fit the target location using the information obtained from the previously extracted key points. The adaptive patch stitching process is presented in Section 4.3.1.

- **Keypoint matches sorting**: the order by which the matching keypoints are considered in the $RD_V D_M$ optimization process, which selects the best keypoint matches, is critical for the final performance, both in terms of visualization and searching. Thus, the keypoint matches are sorted according to their MSE reduction potential before proceeding to the keypoint matches selection. The matching keypoints sorting process is presented in detail in Section 4.3.2.

- **Joint $RD_V D_M$ optimization keypoint matches selection**: the most critical step in the proposed video coding solution is the parsimonious selection of the set of keypoint matches that will be most beneficial for the given objective in terms of visualization (visual quality performance, D_V) and searching (descriptor matching performance, D_M). Due the dual-purpose nature of the proposed coding solution, ideally the selected keypoint matches should offer a performance trade-off both in terms of reducing the visual quality distortion as well as the descriptor matching distortion. This is so because giving privilege to one distortion may likely penalize the other. In the designed framework, the balance between the two functionalities may be adjusted depending on the application scenarios requirements, notably to the extreme cases where the visualization or searching capabilities have total predominance. In Section 4.3.3, the matching keypoint selection problem is tackled in detail using a general framework where the rate, visual quality distortion and descriptor matching distortion may be jointly optimized by appropriately weighting the relevance of the two distortions.

- **Selected keypoint matches coding**: for efficiency, the parameters of the selected keypoints, notably position (x, y), scale σ and angle θ, are differentially coded with respect to the corresponding matching key points from the reference frames. Before entropy coding, the residue for each keypoint parameter above is scalar quantized to reduce its coding rate. An adaptive arithmetic encoder is used to entropy code the various syntactic elements.

- **Base layer coding**: with the selection of the optimal keypoint matches following the defined visualization-searching performance trade-off, the base layer coding process is completed. In summary, the base layer output bitstream comprises the k-frames coding bitstream, generated using a standard video codec, and the f-frames coding bitstream, corresponding to the set of selected keypoint matches. These selected keypoint matches should provide frame correspondences, which may be both efficient in terms of visualization by improving the interpolated f-frame visual quality and in terms of searching by increasing the number of descriptor matches.

- **Enhancement layer residue coding**: although the base layer solution might be the appropriate fit for networks with very strict bandwidth constraints, it may be desirable for other application scenarios to be able to further improve the reconstructed f-frames quality. In fact, as the Base Layer (BL) does not code any texture residue for the f-frames, its quality is limited by the novelty that can migrate from the decoded reference frames, typically preventing it to achieve very high quality. In this context, by coding the residue between the original f-frame and the corresponding reconstructed BL f-frame, the Enhancement Layer (EL) is able to add detail and quality to the BL, naturally at the cost of some additional rate. The residue is coded by applying a conventional transform-quantization scheme, in this case an HEVC-like coding solution. A straightforward adaptation of the HEVC reference software (HM, version 16.3) was made for residue coding. This enhancement layer makes the proposed coding framework quality scalable for the f-frames, which is a functionality not provided by the HEVC standard.

- **Visualization and matching**: the proposed dual-purpose coding framework aims at delivering both efficient visualization and searching experiences. In this context, the decoding process offers not only a pixel-based reconstruction for visualization but also a set of searching efficient f-frame keypoint positions, extracted from the original video frames (which naturally are not available at the decoder). These (original) keypoint positions may drive now the descriptor extraction process, and consequently the following descriptor matching process, targeting the best matching performance and thus searching experience. Conventional video coding solutions like HEVC have to extract the keypoints from the lossy decoded video, thus obtaining less reliable positions.

4.2.2 Decoder

Since the decoder is mostly embedded in the encoder, decoding proceeds essentially as already described for the encoder. The exception is the decoding of the matching keypoint pairs. In summary, the decoder processes first the k-frames and interpolates the appropriate number of f-frames depending on the GOP size. Then, the f-frames are improved by patch stitching using the reference frames patches defined after decoding the selected keypoint matches whose location is differentially coded relative to the keypoints detected in the reference frames.

As usual, the most critical coding tool in the proposed framework is the one that decides how the rate is spent, in this case the 'clever' encoder selection of the keypoint matches to code. In this type of dual-purpose codec, this selection process is more complex than usual because the visualization and searching performances may have to be jointly optimized depending on the relevant application scenario constraints.

4.3 Coding tools

This section targets the detailed presentation of the most novel and most critical modules in the proposed dual-purpose video coding framework. In this context, before going any further, let us restate the notation. Each visual feature is represented by the pair $\{\mathbf{p}_{n,i}; \mathbf{d}_{n,i}\}$ where $\mathbf{p}_{n,i}$ denotes the vector with the keypoint position (x, y), scale σ and angle θ of the i-th feature in frame n and $\mathbf{d}_{n,i}$ the associated descriptor vector, e.g. SIFT coefficients.

4.3.1 Adaptive patch stitching

The patch stitching process targets to improve the f-frames quality with appropriate patches extracted from the already available (decoded) reference frames. In the stitching process, the image patch $I_m^{(k)}|_{\Omega_{m,j}^{(k)}}$ defined over the region $\Omega_{m,j}^{(k)}$ centered at a selected keypoint location $(x_{m,j}, y_{m,j})^{(k)}$ from a reference frame $I_m^{(k)}$ is extracted and seamlessly stitched over the region $\Omega_{n,i}^{(f)}$ centered at the matching keypoint location $(\hat{x}_{n,i}, \hat{y}_{n,i})^{(f)}$ in the relevant f-frame $I_n^{(f)}$, thus generating the stitched f-frame $I'^{(f)}_n$. The superscripts $^{(k)}$ and $^{(f)}$ refer to the k-frames and f-frames, respectively. In the variables above, the subscripts m and n indicate the m-th and n-th frames, whereas j and i indicate the j-th and i-th keypoints; the hat $\hat{}$ over a variable indicates quantization. For simplicity, circularly-shaped patches are used in this work. The diameters of the image areas involved in the stitching process depend on the scale parameters $\sigma_{m,j}^{(k)}$ and $\hat{\sigma}_{n,i}^{(f)}$ of the matching key points. The reference frame patch diameter is $\mathsf{m}_s\sigma_{m,j}^{(k)}$ and the f-frame destination region diameter is $\mathsf{m}_s\hat{\sigma}_{n,i}^{(f)}$, where

m_s is a scale parameter factor that is adaptively determined for each patch at the encoder as explained in the sequel, and is coded in the bitstream to be used at the decoder.

The stitching process aims to keep unchanged the pixel values both over and outside the boundary $\partial\Omega$ of $\Omega_{n,i}^{(f)}$, while blending inside the pixel values of the patch $I_m^{(k)}|_{\Omega_{m,j}^{(k)}}$ (from the reference frame) seamlessly with those from the f-frame $I_n^{(f)}$. A comprehensive formulation of this problem is given in [27, 109]. In this work, the core patch stitching process is performed using the Poisson stitching technique proposed in [109]. Appendix C provides a fairly straightforward review of the Poisson stitching technique. The patch stitching process is carried out using the non-quantized keypoint parameters of the decoded reference frame and the quantized keypoint parameters of the original f-frame as the second will have to be quantized when coding.

Given the matching keypoints $\hat{\mathbf{p}}_{n,i}$ and $\mathbf{p}_{m,j}$, the first one in the current f-frame (reconstructed up to this point) and the second in the most similar frame found in the reference buffer together with the corresponding frames $I_n^{(f)}$ and $I_m^{(k)}$, the stitching process proceeds as follows (the index n refers to the f-frame, the index m to the reference frame and a hat over a variable indicates quantization):

1. **Initialization**: set the current visual quality minimum distortion $D_{V,cur}$ equal to the distortion between the current f-frame and the corresponding original f-frame. The MSE is used here as visual quality distortion.

2. **Support size adaptation**: for each m_s value in the selected range do:

   Let $I_m^{(k)}|_{\Omega_{m,j}^{(k)}}$ be the image patch defined over a circularly-shaped domain $\Omega_{m,j}^{(k)}$ defined by its diameter $\mathsf{m}_s\sigma_{m,j}^{(k)}$ and centered at the keypoint location $(x_{m,j}, y_{m,j})^{(k)}$ and $\Omega_{n,i}^{(f)}$ the destination region in the f-frame centered at the keypoint location $(\hat{x}_{n,i}, \hat{y}_{n,i})^{(f)}$ with diameter $\mathsf{m}_s\hat{\sigma}_{n,i}^{(f)}$.

   a. **Geometric transform**:

      i. Rotate the reference patch by $\varphi = \hat{\theta}_{n,i}^{(f)} - \theta_{m,j}^{(k)}$ and scale it by $s = \frac{\hat{\sigma}_{n,i}^{(f)}}{\sigma_{m,j}^{(k)}}$ around the point $(x_{m,j}, y_{m,j})^{(k)}$ applying the transformation:

$$A = \begin{bmatrix} \alpha & \beta & (1-\alpha)x_{m,j}^{(k)} - \beta y_{m,j}^{(k)} \\ -\beta & \alpha & \beta x_{m,j}^{(k)} + (1-\alpha)y_{m,j}^{(k)} \end{bmatrix} \tag{4.1}$$

      where $\alpha = s \cdot \cos\varphi$ and $\beta = s \cdot \sin\varphi$

      ii. Translate the reference patch to the appropriate f-frame position by applying:

$$T = \begin{bmatrix} 1 & 0 & \hat{x}_{n,i}^{(f)} - x_{m,j}^{(k)} \\ 0 & 1 & \hat{y}_{n,i}^{(f)} - y_{m,j}^{(k)} \end{bmatrix} \tag{4.2}$$

b. **Poisson stitching**: carry out the stitching process as described in [109]. Appendix C reviews the Poisson stitching technique used in this work.

c. **Visual quality assessment**: compute the visual quality distortion between the resulting stitched f-frame and the original f-frame for each successive m_s value. If the visual quality distortion is reduced regarding $D_{V,cur}$, $D_{V,cur}$ is updated with the new distortion value and the new best scale parameter factor m_s is adopted.

This process returns the stitched f-frame $I'^{(f)}_n$ with the support size $\mathsf{m}_s\hat{\sigma}^{(f)}_{n,i}$ providing the largest visual quality gain. At the decoder, the patch stitching process does not have to be adaptive as the appropriate m_s value is transmitted by the encoder as side information.

Regarding the Poisson stitching technique used in step b above, it is worth to notice that it comes down to solve a system of linear equations of the form $\mathsf{Kx} = \mathsf{b}$. Since the matrix K is symmetric and positive-definite [109], the iterative method Conjugate Gradient [110–113] has been used for solving the resulting linear systems. The mathematical formulation of the Conjugate Gradient method guarantees convergence in at most n steps [110, 111], in the particular case of the seamless stitching above, n is the number of samples within the stitching region $\Omega^{(f)}_{n,i}$ of the f-frame. Further discussion on Poisson stitching can be found on Appendix C and for a detailed treatment on the convergence analysis of the Conjugate Gradient refer to [111].

4.3.2 Keypoint matches sorting

The order by which the keypoint matches are considered in the joint RD_VD_M optimization process has a significant impact on the final performance, both in terms of visual quality as well as searching performance; therefore, it is essential to previously and appropriately sort the keypoint matches using some appropriate criterion as performing an exhaustive search over all possible keypoint matches arrangements is simply impractical due to the prohibitive computational cost. A reasonable solution is to evaluate each candidate keypoint match independently and sort them using a criterion which is able to express its effectiveness in contributing to reduce the visual quality distortion and the descriptor matching distortion (thus ultimately increasing the number of descriptor matches). Naturally, the quality of the descriptors extracted at the decoder to be used for the matching process strongly depends on the quality of the reconstructed frames. Thus, it is considered here that an appropriate criterion to perform the sorting before proceeding to the next joint RD_VD_M optimization process is the MSE reduction relative to the original f-frame caused

by the refinement of an interpolated f-frame using the image patch associated to a specific keypoint match.

To avoid using in this sorting process the complex Poisson stitching process presented before, the potential MSE reduction for each keypoint match is assessed by simply copying the image patch centered at the keypoint location in the reference frame over the matching keypoint location in the f-frame and computing the difference to the original f-frame. This is a low complexity stitching process which avoids solving the involved Poisson equation [27, 109] at the penalty of obtaining only an estimation of the MSE reduction; this is, however, enough for sorting purposes. Moreover, for complexity reasons, this process is performed for every keypoint match independently, implying that the cumulative effect of the keypoint matches is not considered. At the end, the list will include all the keypoint matches ordered by their MSE reduction potential.

4.3.3 Joint $RD_V D_M$ optimization keypoint matches selection

The proposed dual-purpose video coding framework aims at delivering optimal visual quality for visualization and original keypoint information for searching. As previously outlined, to accomplish such objectives, the proposed coding framework combines the pixel-based and feature-based approaches to represent the k-frames and f-frames arranged in a GOP structure. The periodic k-frames are coded using a standard video codec and are also reused as source of image patches to improve the f-frames. In turn, each f-frame is coded using a feature-based approach on top of a first estimation obtained by motion interpolation using the available reference frames, mostly k-frames.

More specifically, the f-frames are coded resorting to a set $\mathcal{M}_{\text{kp}}$ of keypoint matches, $\hat{\mathbf{p}}_{n,i}^{(f)} \mapsto \mathbf{p}_{m,j}^{(k)}$, where $\hat{\mathbf{p}}_{n,i}^{(f)}$ (the hat over indicates quantization) belongs to the current f-frame and $\mathbf{p}_{m,j}^{(k)}$ to a reference buffer frame, always a k-frame for GOP of size 2.

Such dual-purpose coding framework creates the challenge of allocating the bit budget to those keypoint matches which provide the best trade-off between reducing the visual quality distortion (visualization performance) and increasing the number of correct descriptor matches for the images in a given decoder content database (searching performance).

Measuring the visual quality distortion is straightforward as the availability of the original and decoded f-frames at the encoder facilitates the measurement of the distortion reduction associated to a specific keypoint match. However, the situation is very different for the searching capability, as the descriptor matching performance cannot be precisely measured at the encoder as only the decoder has access to the

target content database. In the sequel, the joint optimization framework and joint keypoint matches selection process will be presented.

(1) *Joint Lagrangian optimization framework*

The optimization goal is to select a set of keypoints matches, $\mathcal{M}_{\text{kp}}$, which minimize the following Lagrangian cost function:

$$\arg\min_{\mathcal{M}^*_{\text{kp}}} J = (D_V + \gamma D_M) + \lambda R(\mathcal{M}_{\text{kp}}) \tag{4.3}$$

where D_V is the visual quality distortion, D_M is descriptor matching distortion and $R(\mathcal{M}_{\text{kp}})$ is the total rate for coding the set of selected keypoint matches. The parameter γ weights the importance given to the searching performance regarding the visualization performance, while λ weights the overall rate regarding the combined distortion. It is worth to notice that the visual quality distortion and the descriptor matching distortion are assumed to be additive in the joint Lagrangian cost function (Equation 4.3). For the purpose of this work, this is a good enough supposition as suggested by experimental results presented in Chapter 5.

An iterative procedure to be presented in the sequel is adopted to determine the best set of keypoint matches $\mathcal{M}_{\text{kp}}$ which minimize the cost function as defined in Equation 4.3. At each iteration, the benefit (cost function reduction) is evaluated in terms of rate, visual quality distortion and descriptor matching distortion.

Rate metric

The rate for coding each candidate keypoint match $\hat{\mathbf{p}}^{(f)}_{n,i} \mapsto \mathbf{p}^{(k)}_{m,j}$ is computed as follows:

$$R = R(r_k) + R(r_m) + R(\mathsf{m}_s) + R(\mathbf{p}^{(f)}_{n,i} - \mathbf{p}^{(k)}_{m,j}) \tag{4.4}$$

where $R(r_k)$ is the rate to code the reference frame index (this serves to signal which of the (two for GOP 2) reference frames in the buffer is used for patch stitching); $R(r_m)$ is the rate to code the keypoint match index in the reference frame (following the known order provided by the extractor); $R(\mathsf{m}_s)$ is the rate to code the selected scale parameter factor, and $R(\mathbf{p}^{(f)}_{n,i} - \mathbf{p}^{(k)}_{m,j})$ is the rate to perform lossy coding of the residuals of the keypoint match parameters.

To reduce the computational complexity associated to the joint optimization step, the total rate is estimated by computing the self-information of each syntactic element according to the probability models as described in Section 4.3.4.

Visual quality distortion metric

The MSE is adopted for visual quality distortion in this work. In order to decide whether or not to select a particular candidate keypoint match for coding, the encoder computes the MSE between the original f-frame and resulting frame after performing the adaptive patch stitching. More precisely,

$$\text{MSE} = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(I_O^{(f)}(i,j) - I'^{(f)}(i,j))^2 \tag{4.5}$$

where W denotes the width and H the height of the frame in number of samples. $I_O^{(f)}$ denotes the original f-frame and $I'^{(f)}$ the just resulting stitched f-frame.

Descriptor matching distortion estimation metric

The descriptor matching distortion D_M should provide an objective way to assess the contribution of each candidate keypoint match to the searching performance. In this context, the encoder should ideally only spend rate on those keypoint matches likely to produce correct descriptor matches at the decoder side. As only the decoder has access to the target content database, the descriptor matching performance cannot be accurately measured at the encoder. Thus, it is proposed here to estimate this performance at the encoder by mimicking in the best possible way the descriptor matching steps that are performed at the decoder. Such descriptor matching performance estimation enables to formulate a joint Lagrangian optimization [1, 2, 114] framework as defined in Equation 4.3 to trade-off the rate against the joint visual quality and descriptor matching distortion.

More precisely, it is proposed to estimate the searching performance based on the number of matches between the descriptors extracted from the reconstructed f-frames (at keypoint positions to be selected) and those extracted from the original f-frames (at keypoint positions detected at original f-frame), somehow assuming that the database includes a frame rather similar to the original f-frame. For a reliable searching distortion estimation, each candidate descriptor match should satisfy both the ratio test [7, 8] and the symmetric match criterion as it is reasonable to adopt at the encoder the same criterion usually adopted for performing the searching at the decoder. As discussed in Section 3.3.2, the ratio test criterion discards matches whose the ratio between the distance of the closest descriptor and the distance of the second-closest descriptor is greater than a 0.8, whereas the symmetric match (cross matching) criterion assume as correct matches those descriptor matches that occur in both directions (see Section 3.3.2).

The proposed estimator for the descriptor matching distortion is simply defined in terms of the difference between the number of extracted descriptors (256 being

the maximum in our framework as this has been considered enough) and the number of correctly matched descriptors. Figure 4.2 presents the procedure associated to the encoder estimation of the descriptor matching distortion.

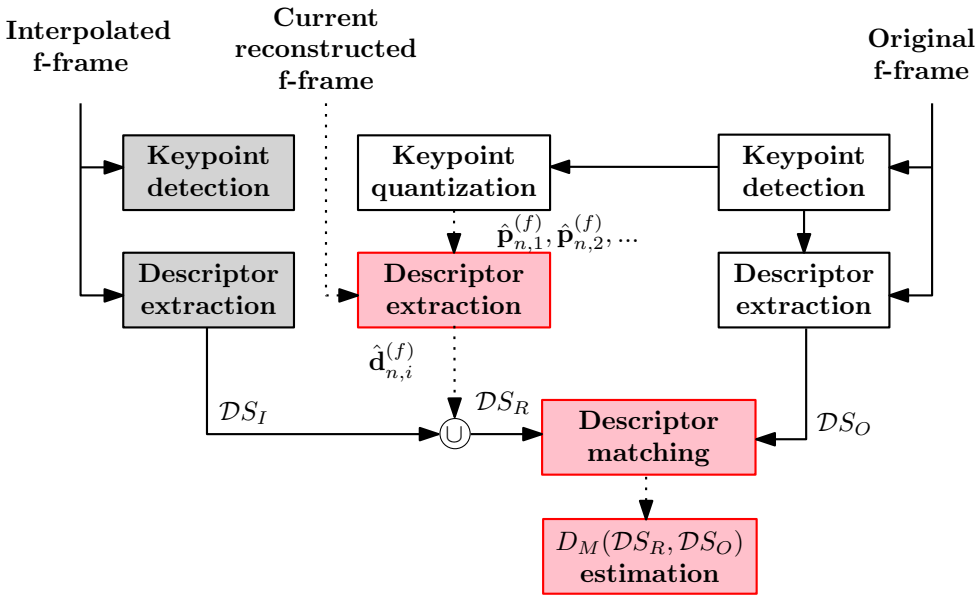


Figure 4.2: Encoder estimation of the descriptor matching distortion. The dashed arrows indicate the iterative steps.

In detail, the descriptor matching distortion estimation proceeds as follows:

1. **Initial descriptor matching estimation**: at the beginning of the joint optimization process, the reconstructed f-frame, $I_R^{(f)}$, is equal to the interpolated f-frame, $I_I^{(f)}$, and thus an initial descriptor matching distortion estimation may be performed using only the set of descriptors extracted from the interpolated f-frame, here labeled as $\mathcal{DS}_I$ (in this case $\mathcal{DS}_R = \mathcal{DS}_I$ as the reconstructed f-frame is the interpolated f-frame). As at this stage there are still no keypoint matches selected, the initial descriptor matching distortion estimation proceeds as follows (extreme left and right branches in Figure 4.2):

    a. **Original f-frame keypoint detection and descriptor extraction**: let $\mathbf{d}_{n,i}^{(f)} = \Psi(\mathbf{p}_{n,i}^{(f)} | I_O^{(f)})$ be a descriptor extracted at keypoint $\mathbf{p}_{n,i}^{(f)}$ detected in the original f-frame $I_O^{(f)}$ and $\mathcal{DS}_O$ the set of such descriptors.

    b. **Interpolated f-frame keypoint detection and descriptor extraction**: let $\bar{\mathbf{d}}_{k,i}^{(f)} = \Psi(\bar{\mathbf{p}}_{k,i}^{(f)} | I_I^{(f)})$ descriptor extracted at the keypoint $\bar{\mathbf{p}}_{k,i}^{(f)}$ detected in the interpolated f-frame $I_I^{(f)}$ and $\mathcal{DS}_I$ the set of such descriptors.

    c. **Descriptor matching for original versus interpolated f-frames**: perform descriptor matching between the original f-frame descriptor set,

$\mathcal{DS}_O$, and the interpolated f-frame descriptor set, as $\mathcal{DS}_R = \mathcal{DS}_I$.

d. **Descriptor matching distortion estimation**: estimate the descriptor matching distortion between the original and interpolated f-frames descriptor sets according to:

$$D_M(\mathcal{DS}_R, \mathcal{DS}_O) = 1 - \frac{\sum_{i=1}^{|\mathcal{DS}_R|} \sum_{j=1}^{|\mathcal{DS}_O|} M_{\mathcal{DS}_R \to \mathcal{DS}_O}(\hat{\mathbf{d}}_i, \mathbf{d}_j) M_{\mathcal{DS}_O \to \mathcal{DS}_R}(\mathbf{d}_j, \hat{\mathbf{d}}_i)}{|\mathcal{DS}_R|} \quad (4.6)$$

where $M_{\mathcal{X} \to \mathcal{Y}}$ is defined as:

$$M_{\mathcal{X} \to \mathcal{Y}}(\hat{\mathbf{d}}_i, \mathbf{d}_j) =$$

$$\begin{cases} 1, \text{if } \forall k \neq j \neq j' \Rightarrow \|\hat{\mathbf{d}}_i - \mathbf{d}_j\|_2 < \|\hat{\mathbf{d}}_i - \mathbf{d}_{j'}\|_2 < \|\hat{\mathbf{d}}_i - \mathbf{d}_k\|_2 \text{ and } \dfrac{\|\hat{\mathbf{d}}_i - \mathbf{d}_j\|_2}{\|\hat{\mathbf{d}}_i - \mathbf{d}_{j'}\|_2} < 0.8 \\ 0, \text{ otherwise} \end{cases}$$

where $|\cdot|$ means cardinality and $\|\cdot\|_2$ is the Euclidean distance. Equation 4.6 measures the fraction of descriptors extracted from the interpolated f-frame not finding a proper descriptor match at the original f-frame descriptor set. It is therefore a descriptor matching distortion. This fraction counts the proportion of descriptors not meeting the ratio test and symmetric matching criteria as expressed by the product $M_{\mathcal{X} \to \mathcal{Y}}(\hat{\mathbf{d}}_i, \mathbf{d}_j) M_{\mathcal{Y} \to \mathcal{X}}(\mathbf{d}_j, \hat{\mathbf{d}}_i)$.

2. **Iterative descriptor matching estimation within the joint Lagrangian optimization**: as the joint optimization process iterates over the sorted keypoint matches, each candidate keypoint match is evaluated regarding the descriptor matching distortion, D_M. Naturally, at this stage, the reconstructed f-frame is no longer the interpolated f-frame but rather its improved version with the successively selected patches associated to the successively selected keypoint matches. The iterative descriptor matching distortion estimation proceeds as follows (central and right branches in Figure 4.2):

   a. **Keypoint parameters quantization**: let $\hat{\mathbf{p}}_{n,i}^{(f)} \mapsto \mathbf{p}_{m,j}^{(k)}$ be the specific candidate keypoint match under consideration in the Lagrangian optimization; the quantized version of all keypoint parameters is considered as the decoder receives $\hat{\mathbf{p}}_{n,i}^{(f)}$ after quantization of all the residues computed for the keypoint parameters relative to the matching keypoint in the relevant reference frame (see Section 4.3.4).

b. **Descriptor extraction at the quantized keypoint**: let $\hat{\mathbf{d}}_{n,i}^{(f)} = \Psi(\hat{\mathbf{p}}_{n,i}^{(f)}|I_R^{(f)})$ be the descriptor extracted in the current reconstructed f-frame (already improved with all the previously selected keypoint matches) at the quantized keypoint position.

c. **Descriptor matching for original versus current reconstructed f-frames**: add the just extracted descriptor $\hat{\mathbf{d}}_{n,i}^{(f)}$ to the set $\mathcal{DS}_R$ and match $\mathcal{DS}_R$ to the descriptor set $\mathcal{DS}_O$ already extracted from the original f-frame.

d. **Descriptor matching distortion estimation**: estimate the descriptor matching distortion between $\mathcal{DS}_R$ and $\mathcal{DS}_O$ as defined in Equation 4.6. Here the descriptors to be matched comprise $\mathcal{DS}_I$ together with those in $\mathcal{DS}_R$, corresponding to the keypoints selected for coding so far. This descriptor matching distortion measures the fraction of used descriptors which did not result into a positive match.

The descriptor matching distortion estimation is performed for each candidate keypoint match, and feeds the joint Lagrangian optimization process that selects the keypoint match based also on the visual quality distortion and the rate, as described in the next section. Figure 4.3 shows the scatter plot of the D_M estimate computed at the encoder side (horizontal axis) and the actual D_M (vertical axis) computed at the decoder side using the target content database. One may notice that the proposed estimation is quite correlated with the actual descriptor matching distortion.

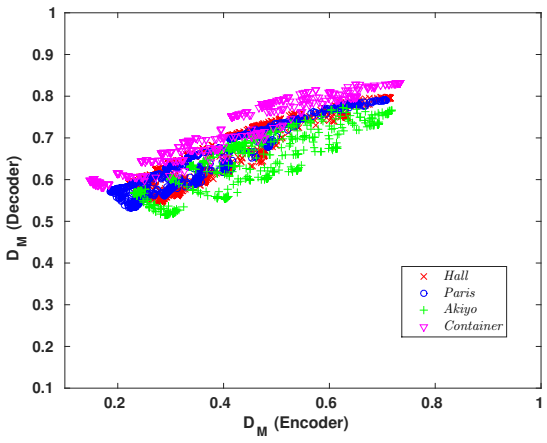


Figure 4.3: Scatter plot of the D_M estimate computed at the encoder side and the actual D_M computed at the decoder side.

(2) *Joint Lagrangian optimization process*

At this point, with the joint optimization framework and metrics properly defined, it is time to design the joint $RD_V D_M$ optimization process to select the optimal keypoint matches.

To determine the final set of keypoint matches to be coded, $\mathcal{M}_{\text{kp}}$, an iterative procedure considering all the available keypoint matches is adopted as follows:

1. **Initialization**: given the selected values for γ and λ, set $\mathcal{M}_{\text{kp}} = \{\emptyset\}$ and initialize the minimum Lagrangian cost function as $J_{min} = (D_{V,ini} + \gamma D_{M,ini}) + \lambda R(\mathcal{M}_{\text{kp}})$. Here $D_{V,ini}$, the initial visual quality distortion, is defined as the MSE between the initially interpolated f-frame and the original f-frame, as at the beginning the reconstructed f-frame is equal to the interpolated f-frame. Also here, $D_{M,ini}$, the initial descriptor matching distortion, is defined as the descriptor matching distortion between the descriptors extracted from the interpolated f-frame, $\mathcal{DS}_I$, and from the original f-frame, $\mathcal{DS}_O$, since at this point $\mathcal{DS}_R = \mathcal{DS}_I$. Lastly, since $\mathcal{M}_{\text{kp}} = \{\emptyset\}$, the rate is naturally zero.

2. **Iterative joint Lagrangian cost reduction**: for each candidate keypoint match in the available sorted list, temporarily add it to the set of selected keypoint matches, $\mathcal{M}_{\text{kp}}$, and evaluate its effectiveness in reducing the Lagrangian cost computed up to the current point. To do so it is needed:

    a. **Rate computation**: compute the accumulated rate for the current set of keypoint matches in the set $\mathcal{M}_{\text{kp}}$ as detailed in Equation 4.4.

    b. **Visual quality performance impact assessment**: to check the visual quality benefit of additionally selecting the current keypoint match, and thus its associated patch, perform the adaptive patch stitching as described in Section 4.3.1 over the current reconstructed f-frame. Then compute the visual quality distortion D_V between the resulting stitched f-frame and the original f-frame.

    c. **Descriptor matching performance impact assessment**: to check the benefit of additionally selecting the current keypoint match, temporarily add the corresponding candidate descriptor to the selected descriptor set $\mathcal{DS}_R$ of the reconstructed f-frame and estimate the descriptor matching distortion D_M using Equation 4.6, notably after matching the descriptor set of current reconstructed f-frame to those of the original f-frame.

    d. **Lagrangian cost computation**: using the rate, visual quality and descriptor matching distortions computed in a, b and c above, compute

the Lagrangian cost $J = (D_V + \gamma D_M) + \lambda R(\mathcal{M}_{\text{kp}})$. If the Lagrangian cost is reduced relative to the current minimum Lagrangian cost, keep this candidate keypoint match in $\mathcal{M}_{\text{kp}}$ (and thus also its stitched patch in the updated reconstructed f-frame), keep its descriptor in $\mathcal{DS}_R$, and update the Lagrangian cost function minimum with this new minimum cost value. Otherwise, discard the keypoint match and its associated descriptor and process the next keypoint in the sorted list.

The above described keypoint matches selection procedure is able to consistently and jointly optimize the visualization and searching performances. The trade-off between visualization and searching distortions minimization depends on the specific application scenario, and can be set by appropriately tuning the Lagrangian cost parameters, λ and γ.

(3) *Lagrangian cost parameters selection*

Naturally, the optimization control parameters γ and λ play a central role in the definition of the optimal configurations using the proposed video coding solution as different trade-offs between the optimization goals can be reached by adjusting them. In addition to γ and λ, another key control parameter is the Quantization Parameter (QP) value used to code the k-frames. In order to properly select the trio of parameters $(\text{QP}, \gamma, \lambda)$ corresponding to various optimal operational points, extensive experiments have been performed as described next. In brief, the joint Lagrangian optimization process presented above was performed for multiple combinations of the parameters $(\text{QP}, \gamma, \lambda)$, thus obtaining a dense cloud of RD_VD_M functional points. Those $(\text{QP}, \gamma, \lambda)$ parameter sets corresponding to RD_VD_M points lying on the convex hull of this dense cloud are selected as providing the best parameter choices. More specifically, this parameter selection process proceeds as follows:

1. RD_VD_M **space filling**: run the coding solution for multiple combinations of the parameter set $(\text{QP}, \gamma, \lambda)$ in some adopted dynamic range for each parameter. Let $config_w = (R, D_V, D_M, \text{QP}, \gamma, \lambda)_w$ be each individual configuration vector including the resulting rate, visual quality distortion and descriptor matching distortion for a particular choice of the input parameter set $(\text{QP}, \gamma, \lambda)_w$ and the parameter set itself. Let $CONFIG$ be the full set of such $config_w$ configuration vectors.

2. RD_VD_M **convex hull creation**: to find the set of RD_VD_M points from $CONFIG$ lying on the convex hull, the widely used convex hull algorithm Quickhull [115] has been used. It gives as output the facets of the convex envelope, that is, the smallest convex set of RD_VD_M points involving the

input set of points. As the objective is here to find the parameter choices $(\mathrm{QP}, \gamma, \lambda)$ which give the optimal $RD_V D_M$ trade-offs, only the lowest facets are kept, this means, those facets which do not have any point below them.

Figure 4.4 shows an example with the full cloud of $RD_V D_M$ points (red) and the Delaunay triangulation for the $RD_V D_M$ points lying on the convex hull (blue) for the video sequence *Paris*. In summary, this process where the $RD_V D_M$ points on the convex hull are defined, allows to identify the $(\mathrm{QP}, \gamma, \lambda)$ combinations providing the optimal visualization-searching performances trade-offs. In fact, a whole convex surface of optimal trade-offs can be found. Figure 4.4 shows an example of this convex surface obtained by performing Delaunay triangulation of the convex hull points.

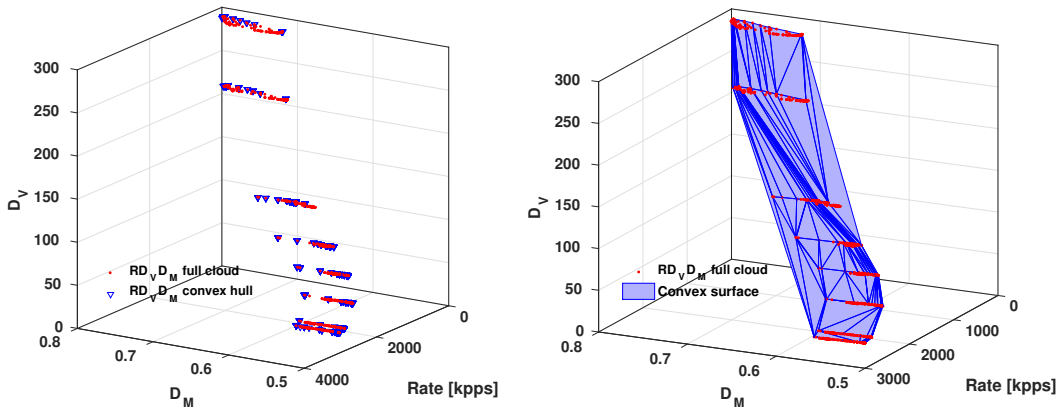


Figure 4.4: Left) Example of the full cloud of $RD_V D_M$ points (red) with the convex hull $RD_V D_M$ points highlighted (blue); right) the corresponding convex surface for the sequence *Paris*.

In order to find these convex surfaces that pass through the $RD_V D_M$ points on the convex hull and therefore finding the appropriate choices for the parameters $(\mathrm{QP}, \gamma, \lambda)$, one needs to perform exhaustive experiments for several parameter combinations. Naturally, this approach guarantees the optimal performance, but it is time-consuming and content-dependent. An analytical relation for the parameters $(\mathrm{QP}, \gamma, \lambda)$ has been searched for by fitting a curve to a set of experimental results with the objective of reducing drastically the computation time at the cost of some performance loss. The procedure described above has been carried out for a few training sequences in order to find the appropriate choice for the parameters $(\mathrm{QP}, \gamma, \lambda)$ and the resulting parameter choices (corresponding to those $RD_V D_M$ points on the convex hull) were used as training data to fit a function. However, the first results arrived at a function which is still content-dependent, therefore, more work is needed on this.

4.3.4 Selected keypoint matches coding

In the proposed Dual-Purpose Video Coding solution, a set of keypoint matches is selected to code each f-frame at the encoder side. In terms of visualization, these selected keypoint matches indicate texture patches from reference frames which are worthwhile to be reused to improve regions in the interpolated f-frame most needing quality improvement. In terms of searching, the selected keypoint matches indicate image positions within the f-frame worth extracting feature descriptors as they are highly expressive is terms of searching.

In this context, to replicate the encoder patch stitching process and to indicate where to extract the descriptors at the decoder side, for each selected keypoint match, the following syntactic elements are coded: a) index of the reference frame in the reference frames buffer, r_k, e.g. previous or next for GOP size 2; b) index r_m of the matching keypoint in the reference frame available at the decoder considering the order given by the keypoint detector itself; c) encoder selected multiplicative scale factor, m_s; d) quantization of the residue $\mathbf{p}_{n,i}^{(f)} - \mathbf{p}_{m,j}^{(k)}$ for each keypoint parameter set, notably the residues for the keypoint parameters position, angle and scale. Note that these are residually coded using as reference the corresponding elements in the matching reference frame keypoint in order to exploit their inter-frame redundancy. In addition, to further reduce the rate, these residues are scalar quantized. This is detailed in the sequel.

(1) *Position residue and angle residue quantization*

Both the position residue and the angle residue are quantized applying the same quantization scheme. For instance, the angle parameter residue for each matching keypoints pair, $c_\theta = \theta_{n,i}^{(f)} - \theta_{m,j}^{(k)}$, is quantized as:

$$\text{round}\left(\frac{\theta_{n,i}^{(f)} - \theta_{m,j}^{(k)}}{QS}\right) QS \tag{4.7}$$

where $\theta_{n,i}^{(f)}$ is the angle parameter of the i-th keypoint in the n-th f-frame, similarly $\theta_{m,j}^{(k)}$ for the matching keypoint in the k-frame, QS is the quantization step. Thus, the f-frame decoded keypoint angle parameter is given by:

$$\hat{\theta}_{n,i}^{(f)} = \theta_{m,j}^{(k)} + \text{round}\left(\frac{\theta_{n,i}^{(f)} - \theta_{m,j}^{(k)}}{QS}\right) QS \tag{4.8}$$

where $\theta_{m,j}^{(k)}$ for the keypoint in the reference frame is not quantized as it is extracted at the decoder.

The same procedure is carried out for the position residue quantization. In this

work, the quantization is carried out using a common quantization step, $QS = 0.25$, which has been validated with exhaustive experimentation.

(2) *Scale residue quantization*

As for the scale quantization, there is one more step as the scale parameter depends on the integer parameters octave (o) and layer (l) according to:

$$\sigma = \sigma_0 \cdot 2^{(o+\frac{l}{3}+\Delta\sigma)} \tag{4.9}$$

where $\sigma_0 = 1.6$ and $\Delta\sigma$ is the scale offset, which resulted from the SIFT scale refinement [7, 8]. Such parameters are also required for proper descriptor extraction at the decoder side. A differential scheme is used to code the octave and layer with respect to the octave and layer of the matching keypoint in the reference frame. No quantization is applied to the octave and layer residues in order to enable a proper descriptor extraction at the decoder side. Then, the scale residue between the 'true' scale value and its approximation computed using the octave and layer values is quantized as follows:

$$\text{round}\left(\frac{\sigma_0\left(2^{(o+\frac{l}{3}+\Delta\sigma)} - 2^{(o+\frac{l}{3})}\right)}{QS}\right) QS \tag{4.10}$$

The same $QS = 0.25$ as above is used for scale residue quantization.

(3) *Entropy coding*

For better compression efficiency, the syntactic elements r_k, r_m and m_s are coded using arithmetic coding [63] with adaptive probability models, initialized with uniform probabilities. On the other hand, the keypoint parameter residues are coded using adaptive arithmetic coding with an initial statistical model set up for each parameter. The initial statistical models are obtained from a set of training sequences different from the set of test sequences. One can roughly estimate the rate associated to each syntactic element by considering a coding set up using 2 reference frames, a maximum of 256 keypoints per frame, 16 multiplicative scale factor values, CIF resolution and a maximum scale residue value of 80. In the worst case, using a uniform probability model for each syntactic element, the coding of each keypoint match would require 1 bit for r_k, 8 bits for r_m, 4 bits for m_s, 23 bits for the position residue, 12 bits for the angle parameter residue, 9 bits for the scale parameter residue, and 5 bits for the octave and layer, in a total of 62 bits per keypoint match. As it is proposed to use entropy coding with adaptive probability models,

this rate can be reduced 1.6 times approximately. Figure 4.5 shows the average rate expenditure for coding each syntax element of the keypoint match selected by the joint Lagrangian optimization process.

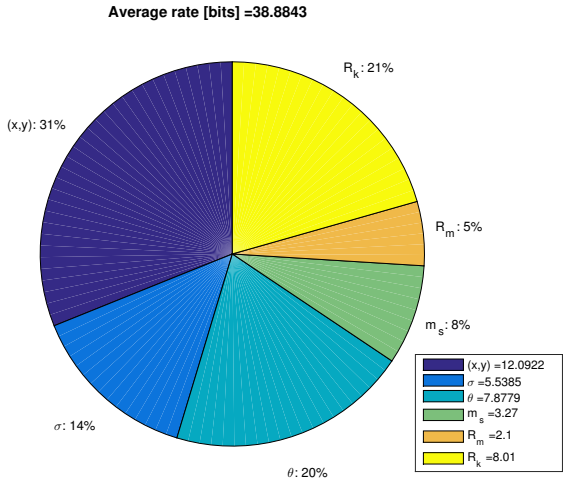


Figure 4.5: Average rate expenditure for each syntax element.

4.3.5 Enhancement layer residue coding

To improve the quality of the Base Layer (BL) reconstructed f-frames with its own novelty (and not only that migrating from the reference frames), a residue is computed between the original f-frame and the corresponding reconstructed BL f-frame. This residue is coded with an HEVC-like coding solution where the reconstructed BL f-frame plays the role of the HEVC prediction and the HEVC transform and quantization and entropy coding tools are used to code the Enhancement Layer (EL) residue. The adopted HEVC-like solution was built upon the HEVC reference software (HM version 16.3) [3, 4, 65].

Conceptually, this HEVC-like residue coding process consists in substituting the HEVC prediction module with the proposed BL decoder, which creates its prediction by using the keypoint matches (which behave like motion estimation) and patch stitching on top of an initially interpolated f-frame. The EL residue coding process includes the following steps:

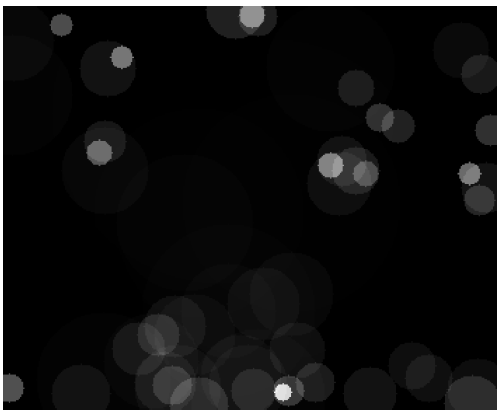
1. **Enhancement layer prediction**: in the HEVC encoder, the residue for each Coding Unit (CU) is obtained after defining one or more Prediction Units (PUs) and subtracting the (Intra or Inter) predicted blocks from the original block. Similarly, in the proposed HEVC-like EL residue coding, the residue for

each CU is obtained by subtracting the reconstructed BL output block from the original block.

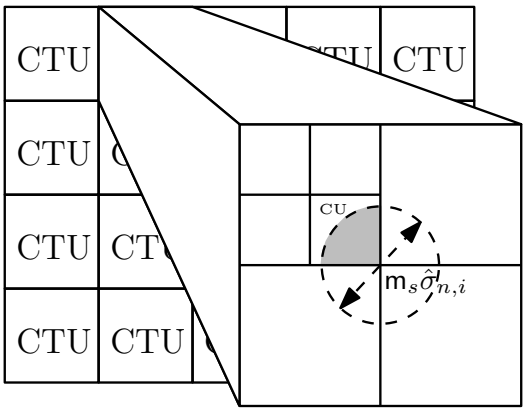
2. **Base layer rate allocation map creation**: in this context, the rate associated to the prediction creation is here the rate used to code all the syntactic elements associated to the coding of the f-frame BL using the matching keypoint pairs. To perform the HEVC-like residue coding at CU level, it is necessary to compute its corresponding rate, in this case its rate share of the f-frame BL. To compute this rate, the BL produces a rate allocation map with an estimation of the BL rate expenditure for each area of the f-frame, notably depending on how the stitching process is distributed within the f-frame. More precisely, the number of bits spent for coding each keypoint match is divided by the number of pixels in the corresponding stitched area. Figure 4.6a shows an example of such rate allocation map where the whiter the area, the higher the rate estimation.

3. **Coding unit rate estimation**: the rate allocation map is used by the HEVC-like residue coding module to estimate the rate already used in the BL prediction. This is so because each CU should consider the rate previously spent by the BL in the corresponding area. Otherwise, the residue coding would be done without taking into account the BL rate. The rate for each CU is estimated by accounting the rate previously spent for the corresponding area by BL rate allocation map as shown in Figure 4.6b. In practice, for each CU, the proposed HEVC-like coding takes as prediction creation rate the rate corresponding to the stitching process parameters for the BL.

4. **Residue coding**: after the prediction and rate estimation steps, the residual block coding occurs as in the HEVC encoder, notably involving the transform, quantization and entropy coding steps. The QP for f-frame residue coding is equal to the QP for k-frames incremented by 1 to implement some amount of quantization cascading.



(a) Rate allocation map for the f-frame BL for frame 73 of the test sequence *Foreman*.



(b) Overlapping example between a Coding Unit (CU) and a stitched BL region (grey area).

4.4 Final remarks

This chapter has described the proposed Dual-Purpose Video Coding Solution, which was designed to address applications that consider visualization and searching needs. The proposed solution is based on a flexible joint Lagrangian optimization framework which combine pixel-based and feature-based processing and is able to appropriately trade-off visualization and searching performances. The Chapter 5 hereafter presents the performance assessment of the proposed video coding solution both in terms of visualization and searching, and compares its performances regarding the state-of-the-art HEVC standard.

This page is intentionally left blank.

Chapter 5

Performance assessment

This chapter presents experimental results for the assessment of the proposed DPVC solution under meaningful test conditions. The results show the flexibility of the proposed coding solution to achieve different optimization trade-offs, notably when allocating the bitrate budget while jointly targeting visualization and searching capabilities. The state-of-the-art HEVC standard will be used as the natural benchmark to compare the obtained performance, considering not only joint optimization objectives but also special cases where the optimization is biased towards visualization or searching.

5.1 Test material and conditions

To appropriately assess the proposed DPVC solution in terms of visualization and searching performances, the following materials and test conditions have been adopted:

- Four surveillance and personal communications video sequences have been selected, notably *Hall*, *Container*, *Paris* and *Akiyo*. All sequences are in CIF spatial resolution at 30Hz and 10 seconds long (300 frames). Appendix B presents the set of used video sequences. The choice of low spatial resolution sequences reflects the video coding scenarios addressed in this research work; this choice does not imply any disadvantage for the adopted benchmark solution. In addition, the proposed solution can readily be run in sequences of high spatial resolution (High Definition (HD), 1280×720). A thorough analysis of the HD case will be the subject of a different publication.

- To measure both the visualization and searching performances in a reliable way, each test sequence has been divided in two halves. To assess the visualization performance, the first half was used for coding. To assess the searching performance, the original version of the last frame of the second half (thus

minimizing the correlation with the coded frames from the first half) was used to build the target content database at the decoder side; these frames play the role of target content for the queries based on decoded video frames.

- The selected QP values for k-frames coding were 45, 40, 37, 34, 30 and 25. The γ values were set in the range 0 and 1000 and the λ values in the range 0 and 1 to accommodate the different distortion scales (see Equation 4.3).

- A maximum number of 256 SIFT descriptors [8] was extracted per frame; for each keypoint, the residue of the parameters position and scale are quantized with a precision of one quarter of pixel while the angle is quantized with a precision of one quarter of degree.

- A GOP size of 2 frames was selected. The reference frames buffer always includes two reference frames (one past and one future); for GOP size 2, these reference frames are the past and future k-frames for each f-frame.

5.2 Benchmarks and metrics

The natural benchmark for the proposed coding solution is the state-of-the-art HEVC standard, notably its reference software HM, version 16.3 [51]. The Main profile has been selected while using two prediction structures: *All Intra* and *IBI*. It is important to stress that this is a very tough benchmark as it represents the best result of the video coding technology evolution designed by the related research community over the past few decades. Comparing a new, naturally less mature, coding solution with such a mature benchmark is by itself a challenge. To perform a solid, wide and meaningful evaluation, the following performance metrics have been adopted:

- **Keypoint extraction performance**: the repeatability score [14] between the keypoints detected in the original f-frames and those extracted from the decoded f-frames is used to evaluate the impact of compression on the quality of the keypoints positioning. This is so because this later impacts the extracted descriptors. The repeatability score is defined as the ratio between the number of keypoint correspondences (see Section 3.2.2) in the two images and the smallest number of detected keypoints in the two images and is averaged over all f-frames.

- **Visual quality distortion**: the usual MSE is adopted as the visual quality distortion metric to evaluate the performance regarding visualization. Also, the Bjontegaard-Delta metrics [75], notably the BD-Rate is used to compare

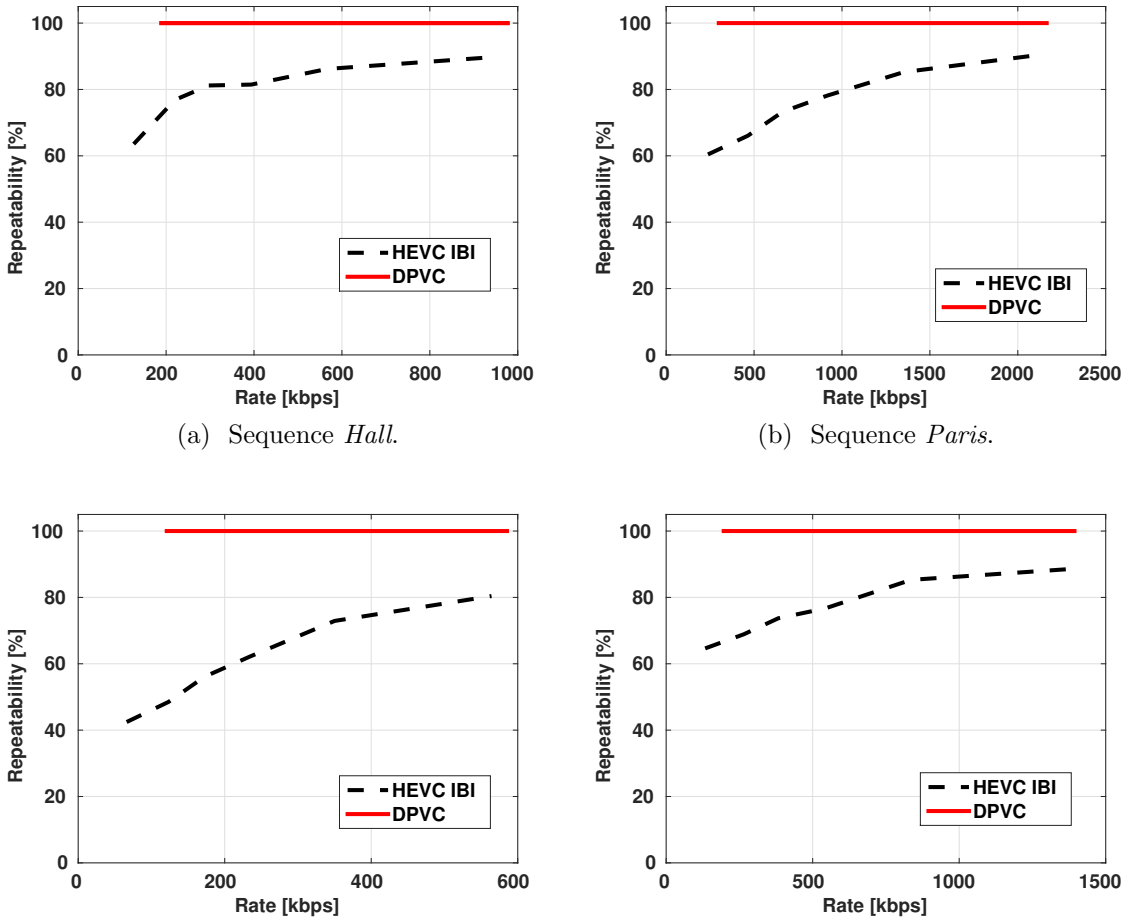
alternative coding solutions in terms of RD performance, that is, rate reduction for equivalent quality. The visual quality distortion assessment considers all coded frames, both f-frames and k-frames, as these frames types are not independent from each other.

- **Descriptor matching distortion**: the searching performance is evaluated by the average, computed over all decoded frames, of the fraction of descriptors extracted from the decoded video (query descriptors) which positively match the descriptors extracted for each image in the target content database. These positive descriptor matches must satisfy both the ratio test and the symmetric matching criteria to be declared proper, positive matches. The descriptor matching distortion is the complementary fraction of the descriptor matching performance as defined in Equation 4.6. Notice that here the 'true' descriptor matching performance is computed (and not an estimation), which may only be assessed at the decoder side with access to the (original) content database. The descriptor matching distortion assessment considers only the f-frames (against the HEVC B frames) as there are no keypoints coded for the k-frames.

5.3 Keypoint repeatability performance

Repeatability is a fundamental property for visual features. Matching performance based on visual features relies on the property of detecting the same distinguishing locations on images depicting the same scene content, although acquired or processed differently. Notably, image and video compression have a detrimental effect on keypoint detection, mainly at lower bitrates where a large quantization step and blocking artifacts may create spurious keypoint responses and erase valuable ones. This in turn would imply extracting descriptors at image locations unlikely to be correctly matched with descriptors extracted from original images. The first main advantage of the proposed DPVC solution is the availability at the decoder of originally extracted keypoint locations what is not possible for the alternative HEVC solution. Fig. 5.1 shows the repeatability score averaged over all f-frames for DPVC and over all B-frames for an HEVC IBI configuration. The proposed DPVC consistently achieves a repeatability score of 100%, meaning that the keypoint locations are essentially the same as obtained from original frames (despite the quantization applied to the keypoint parameter residues). This is essentially different from the HEVC repeatability behavior as the compression process has a significant detrimental effect on the keypoint locations, especially at the lower bitrates where the repeatability score drops significantly. It is worth to reiterate that a high repeatability score is fundamental for matching-based applications as one requires repeatable

image locations from where one may extract descriptors likely to produce correct image region correspondences by performing descriptor matching.



(a) Sequence *Hall*. (b) Sequence *Paris*.

(c) Sequence *Akiyo*. (d) Sequence *Container*.

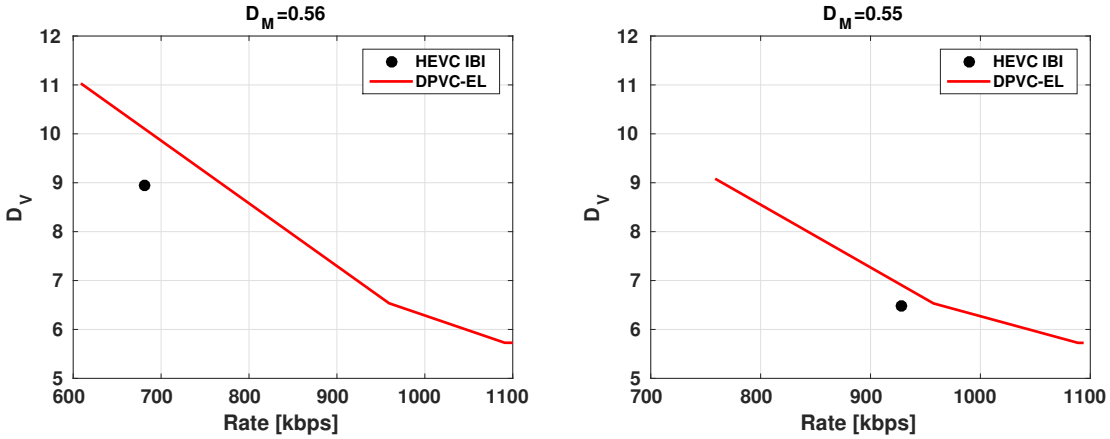
Figure 5.1: Repeatability score averaged over all f-frames/B-frames for the tested sequences.

5.4 Trading-off visualization and searching performances

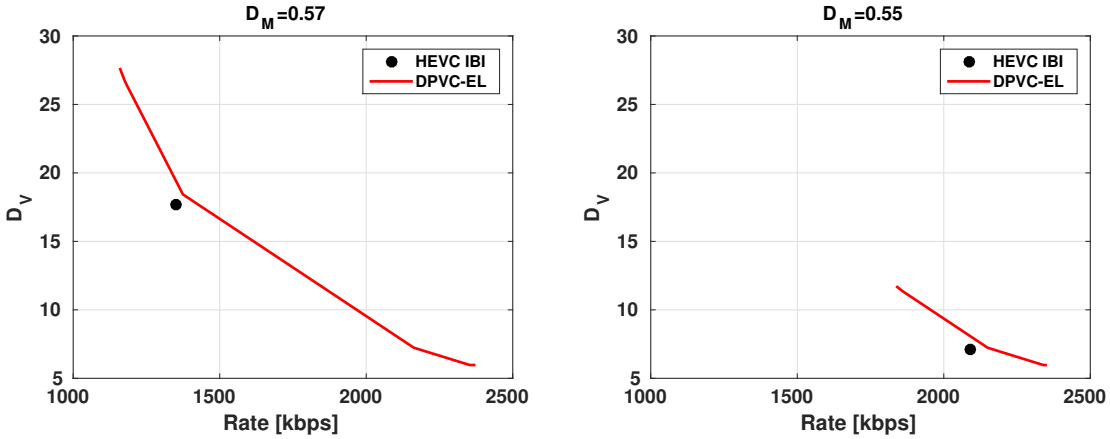
To show the flexibility of the proposed DPVC solution in trading-off visualization (visual quality distortion) and searching performances (descriptor matching distortion) while offering comparable performances regarding HEVC, this section presents and compares RD_V curves for a fixed descriptor matching distortion (D_M). These (level) curves are obtained from the convex surface fitted to the convex hull points as described in Section 4.3.3. While HEVC offers a fixed descriptor matching distortion D_M for a specific RD_V pair, the DPVC solution may offer the same D_M performance

for all the RD_V pairs lying along a curve implying that it is possible to trade-off rate with visual quality without 'touching' the descriptor matching performance. This is a powerful capability that results from the proposed joint optimization strategy and, to our knowledge, only the proposed coding solution can offer.

In Fig. 5.2 and 5.3, the RD_V performance for the proposed DPVC solution is presented for four video sequences at two specific D_M values. As shown, from an RD_V performance perspective, the proposed DPVC solution performs rather similarly to HEVC IBI at the fixed descriptor matching distortion values while offering at the same time many other RD_V combinations for the same matching distortion. The key issue here is that the DPVC solution offers a large set of RD_V operational points for each matching distortion, what is impossible with HEVC. For example, DPVC is able to offer a reasonable increase or reduction in the visual quality distortion by reducing or increasing the bitrate expenditure while keeping fixed the descriptor matching distortion. This behavior evidences that the jointly selected and coded keypoints are effective in holding the descriptor matching distortion at a certain level while trading-off the visual quality distortion. Appendix D.1 presents more extensive results for this RD_V trade-off capability of the proposed solution, notably for additional fixed D_M values.

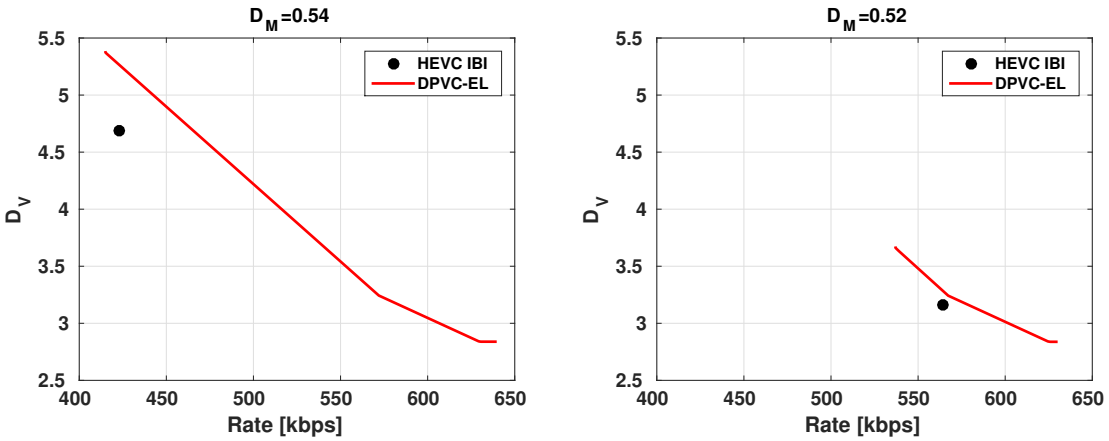


(a) Sequence *Hall*.

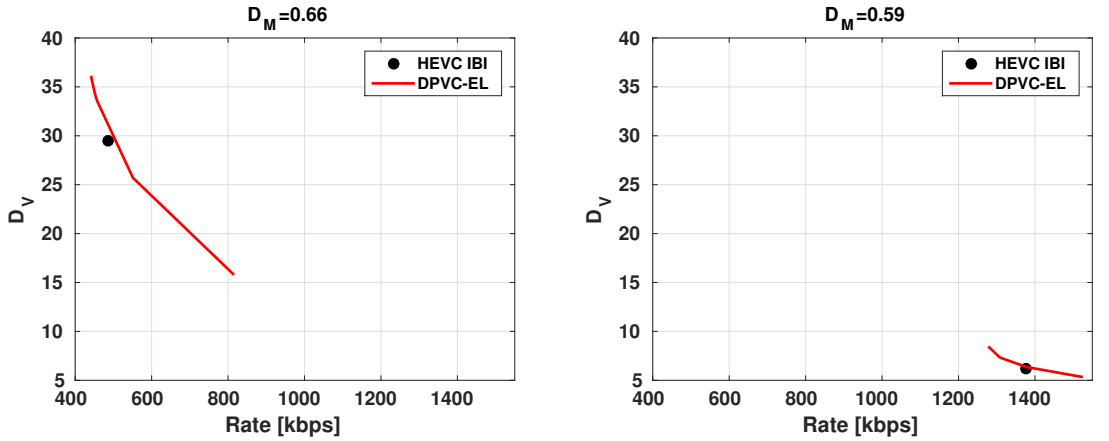


(b) Sequence *Paris*.

Figure 5.2: RD_V performance for two fixed descriptor matching performances for sequences: top) *Hall*; and bottom) *Paris*.



(a) Sequence *Akiyo*.

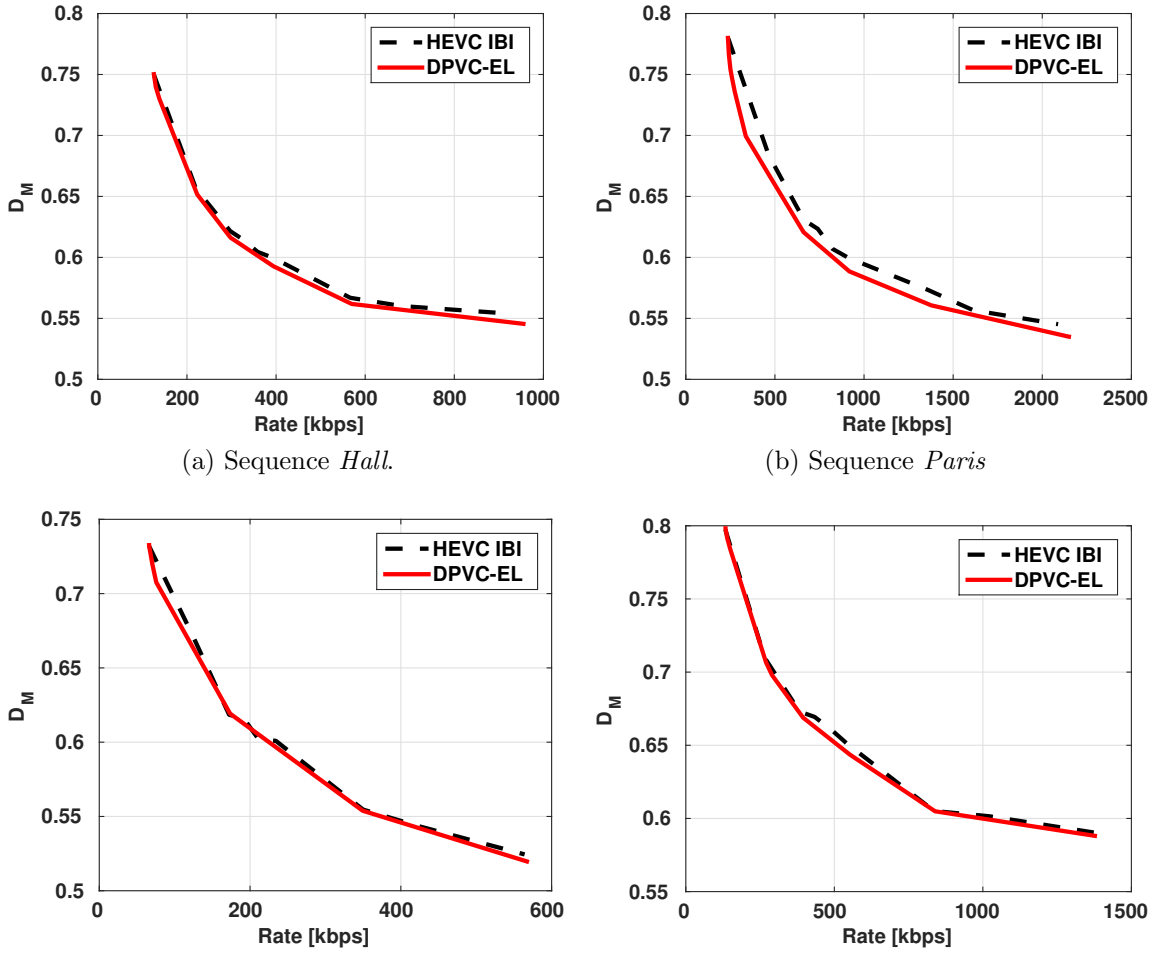


(b) Sequence *Container*.

Figure 5.3: RD_V performance for two fixed descriptor matching performances for sequences: top) *Akiyo* and bottom) *Container*.

5.5 Best searching performance

Among the optimization trade-offs achievable with the proposed DPVC solution is the special case where the operational points are selected to provide the best searching performance. In this jointly optimized video coding framework, this situation is associated to the convex hull points which yield the best trade-off between searching performance and rate. Fig. 5.4 shows RD_M curves expressing the best descriptor matching distortion from the $RD_V D_M$ points on the convex hull. The DPVC solution consistently outperforms the HEVC RD_M performance as it achieves a lower descriptor matching distortion than HEVC for the same bitrate, meaning that the fraction of descriptors not finding a proper match in the target image of the content database is lower for the DPVC. This follows from the fact that, in the proposed DPVC, only a set of keypoints, carefully selected in the joint Lagrangian optimization routine, is conveyed to the decoder so it may extract descriptors at image locations likely to produce correct matches. This again validates the importance of providing reliable keypoint location and consequently descriptor information for improved searching performance. Figure 5.5 shows a sample frame of the coded sequence *Hall* and its target image in the database, for which is superimposed the feature matches produced for three parameters settings. Appendix D.2 provides more details for these operational points selected to provide the best searching performance, notably the actual number of matches as function of the bitrate as well as additional sample frames and the produced feature matches.



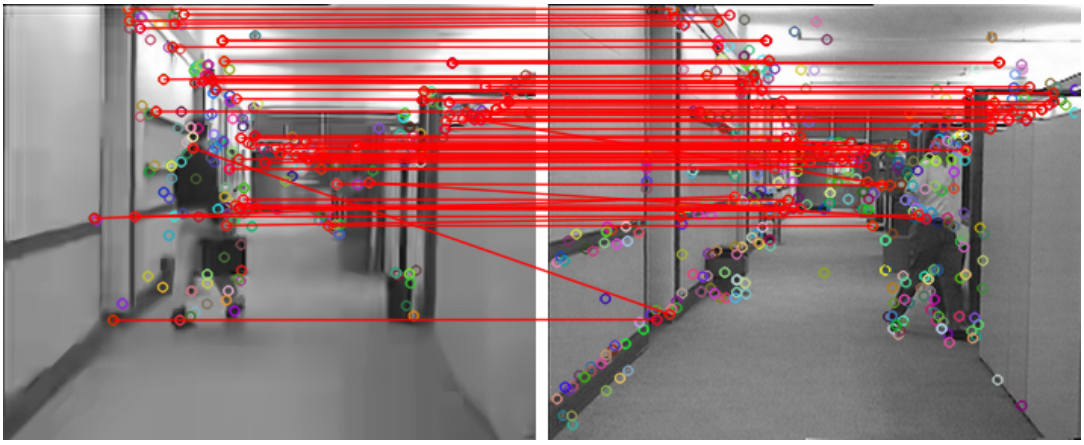
(a) Sequence *Hall*.

(b) Sequence *Paris*

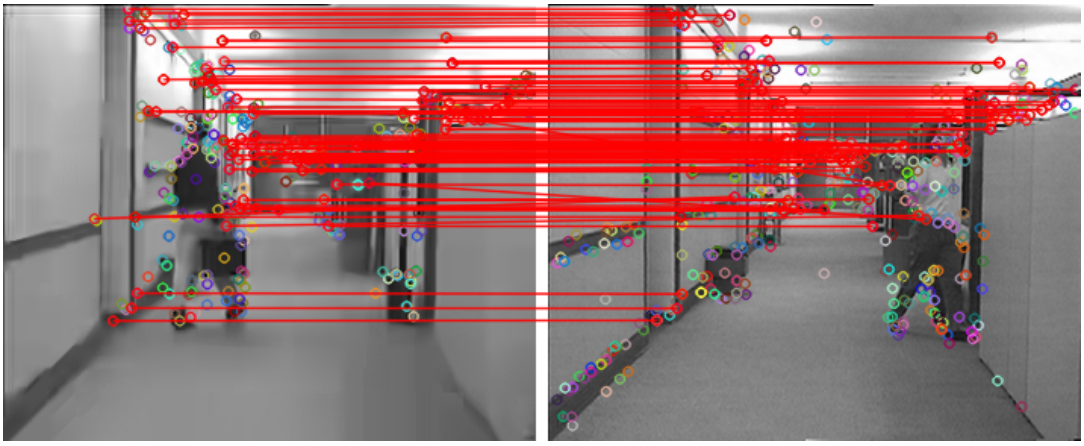
(c) Sequence *Akiyo*

(d) Sequence *Container*

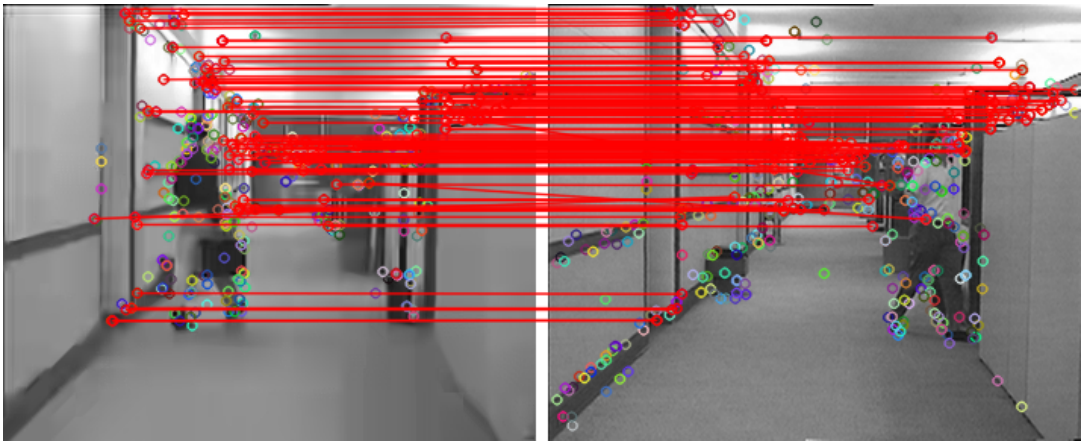
Figure 5.4: Best operational points in terms of RD_M performance obtained from the convex hull points for the tested sequences.



(a) Frame 43, QP = 45, $\lambda = 2^{-5}$ and $\gamma = 50$.



(b) Frame 43, QP = 45, $\lambda = 2^{-10}$ and $\gamma = 50$.



(c) Frame 43, QP = 45, $\lambda = 2^{-20}$ and $\gamma = 50$.

Figure 5.5: Feature matches between a frame of the sequence *Hall* and its reference image in the database, k-frames coded with QP=45.

5.6 Best visualization performance

Another case of special relevance is when the optimization goal is to achieve the best visualization performance; this allows to assess to what extent the proposed coding solution is competitive with the best standard coding solution available in terms of the most commonly used RD_V performance. To this end, the best operational points regarding visualization performance are selected from the convex hull points for the proposed DPVC.

Table 5.1 presents BD-Rate for the proposed DPVC solution regarding HEVC, using the PSNR as visual distortion quality metric instead of D_V. The BD-Rate regarding the HEVC IBI and HEVC All Intra is provided for the for DPVC-BL and DPVC-EL, which correspond to the DPVC base and enhancement layers, and also for the so-called Motion Compensated Frame Interpolation (MCFI) solution where f-frames only result from frame interpolation at no rate cost.

This set of results allows concluding that DPVC-EL performs rather close to HEVC IBI and easily outperforms HEVC All Intra, showing that the use of keypoint matches, which behave like motion vectors, in combination with patch stitching and residue coding do not introduce significant coding performance losses regarding the video coding state-of-the-art as represented by HEVC. In exchange, the proposed solution offers, in a unified fashion, an explicit and flexible coding framework where visualization and searching can be jointly optimized, while still offering good performance for the cases where one optimization target dominates the other. It is important to stress that the obtained BD-Rate loss is typically below 2.5% while offering some amount of quality scalability. When scalability is offered, it is common to accept a BD-Rate penalty up to 10% regarding a meaningful non-scalable solution [116], which is here HEVC as it does not offer any quality scalability. Table 5.1 shows that the penalty is much lower here.

Table 5.1: BD-Rate for DPVC regarding HEVC.

		HEVC All Intra	HEVC IBI
Hall	**DPVC-MCFI**	-42.406%	8.871%
	DPVC-BL	-42.584%	8.578%
	DPVC-EL	-46.957%	**1.600**%
Paris	**DPVC-MCFI**	-40.843%	15.630%
	DPVC-BL	-41.031%	15.243%
	DPVC-EL	-47.599%	**2.416**%
Akiyo	**DPVC-MCFI**	-50.971%	0.948%
	DPVC-BL	-51.070%	0.723%
	DPVC-EL	-50.946%	**0.840**%
Container	**DPVC-MCFI**	-51.769%	1.160%
	DPVC-BL	-51.860%	0.965%
	DPVC-EL	-51.951%	**0.709**%

5.7 Final remarks

This chapter has presented the experimental results for the performance assessment of the proposed DPVC described in Chapter 4. After introducing the used test material and conditions as well as the benchmarks and metrics, the repeatability performance was assessed in Section 5.3, which revealed that the encoder-extracted coded keypoints provided by the DPVC achieve a repeatability score of 100% and that the compression process of the HEVC, mainly at lower bitrates, has a detrimental effect on the repeatability score. As discussed in Section 3.2.2, having at disposal repeatable image locations is fundamental to look for image region correspondences by performing descriptor matching. Section 5.4 has shown the flexibility of the DPVC solution to trade-off rate with visual quality while keeping unaltered the descriptor matching performance. For instance, this implies that the DPVC enables to provide bitrate savings at the cost of a higher visualization distortion while still delivering the same descriptor matching performance. Section 5.5 has presented the best performance in terms of searching needs. For this purpose, the operational points providing the best RD_M performance were selected from the convex hull. The proposed DPVC consistently achieves a lower descriptor matching distortion than HEVC for the same bitrate, meaning that the fraction of descriptors not finding a proper match in the target image of the content database is lower in the DPVC. This is so because only a set of keypoints extracted at the original f-frames, carefully selected in the joint Lagrangian optimization routine, is coded (relative to its matching keypoint in the reference frame) and conveyed to the decoder so it may extract descriptors at image locations likely to produce correct matches. Finally, Section 5.6 has presented and compared the best performance in terms of visualiza-

tion needs, which allow to assess to what extent the DPVC is competitive with the state-of-the-art coding solution in terms RD_V performance, for this purpose the operational points providing the best RD_V performance were selected from the convex hull. The set of results has shown that the proposed pixel-based and feature-based video coding solution does not introduce significant performance losses when compared to HEVC in terms of BD-Rate. In fact, its performance loss is well bellow the 10% limit commonly accepted for scalable solutions [116].

Although not presented in this work, the use of larger GOP sizes increase the relative distance between the f-frames and their reference frames used as source of image patches for improving the interpolated f-frames. As consequence, this likely implies less correlation between f-frames and the reference frames. This tends to impair the coding performance of the base layer which relies on the texture that can be reused from the reference frames. A hierarchical GOP structure can be used to mitigate this performance loss in the base layer, similarly to the hierarchical coding structures for B-frames in the HEVC.

The Chapter 6 hereafter presents the thesis's conclusions and possible further investigations.

This page is intentionally left blank.

Chapter 6

Conclusion and future work

In modern video applications, the role of the decoded video is much more than filling a screen for visualization. Among the emerging required user capabilities, searching plays a key role. In this context, this work proposes a novel Dual-Purpose Video Coding solution that targets not only the usual visualization capabilities but it also potentiates simpler and better searching capabilities by combining the pixel- and feature-based coding approaches. To this end, in order to pave the way for the proposed joint dual-purpose solution, Chapter 2 has presented a review of the main video coding tools underpinning state-of-the-art visualization-driven video coding solutions and also it has reviewed the HEVC standard over which the proposed solution relies on for pixel-based coding approach. Furthermore, Chapter 3 has presented a review of local visual representation for visual content and a study on coding schemes devised to code visual features extracted from video sequences. The objective was to lay down the ground for the feature-based coding approach.

Chapter 4 has presented the proposed video coding architecture that employs a hybrid approach where pixel-based and feature-based coding are jointly used. To this end, the so-called k-frames are coded using pixel-based processing by means of the HEVC and used as reference frames to code the f-frames using the feature-based coding approach. A first estimate of the f-frames is obtained by interpolating them from the neighboring decoded reference frames. Subsequently, the f-frames are first refined by migrating appropriate image patches from the decoded reference frames provided by the selected keypoint matches and then by the HEVC-like residue coding in the enhancement layer. In order to operate this pixel-based and feature-based coding framework considering the dual-purpose objective, a flexible and unified Lagrangian optimization framework has been designed, which explicitly takes into account the rate and the visual quality and descriptor matching distortions. To allow this joint Lagrangian optimization framework, the descriptor matching performance is estimated at encoder side by matching the descriptors extracted at the reconstructed f-frame using the candidate keypoint matches data to those of the original

content. The idea is simply to check if despite coding artifacts the extracted descriptor maintains its distinctiveness and can still be properly matched to its original version. The experimental results show that the proposed solution allows to reach multiple trade-off points in terms of visualization and searching performances with no or only a negligible *RD* performance penalty. Results show that the proposed DPVC solution is able to perform better or very close to the state-of-the-art HEVC IBI solution, if required, while offering increased operational flexibility.

As discussed in chapter 1, a very few research works have been done in the HATC domain to exploit the interaction of the feature-level data, targeting searching, and the pixel-level, targeting visualization. Although this work has made an effort to address this dual-purpose coding video coding scenario, there is still room for different approaches and topics to be considered. For instance, one may investigate a way to relate the parameters λ, γ and QP in order to save computation time without significantly impairing the visualization and searching performance. Future work may also consider the design of a video coding framework where the f-frames are efficiently coded using the descriptors themselves and not only the key point matches. This should allow performing searching not only using original data extracted key points but also using original data extracted descriptors.

Advances in image sensors and comprehensive image modeling by means of the plenoptic function have been pushing forward towards richer representations of the visual information [117, 118], opening up a new range of interesting applications and functionalities. For instance, light field imaging offers functionalities such as change of focus, relighting, change of viewing position and enhanced analysis [117]. In this context, one may say that richer content representations only benefit emerging video-enabled visual analysis tasks (such as visual search). The new video coding solution addressed in this work fits well within this scope.

Appendix A

Published and submitted papers

This appendix presents the list of published and submitted papers resulted from the research work.

A.1 Published papers

C.1 Silva, R. C, Pereira, F., Silva, E. A. B. "Studying the Compression Performance of Video Descriptors". In: Simpósio Brasileiro de Telecomunicações (SBrt), Juiz de Fora, Brazil, September 2015. Recipient of the Best Paper Award of the Symposium.

C.2 Silva, R. C, Pereira, F., Silva, E. A. B. "Feature-based Video Coding: Designing an RD Efficient and Search Friendly Framework". In: Picture Coding Symposium (PCS), Nuremberg, Germany, December 2016.

A.2 Submitted papers

J.1 Silva, R. C, Pereira, F., Silva, E. A. B. "Towards Visualization and Searching: a Dual-Purpose Video Coding Approach", IEEE Transactions on Multimedia.

Appendix B

List of used video sequences

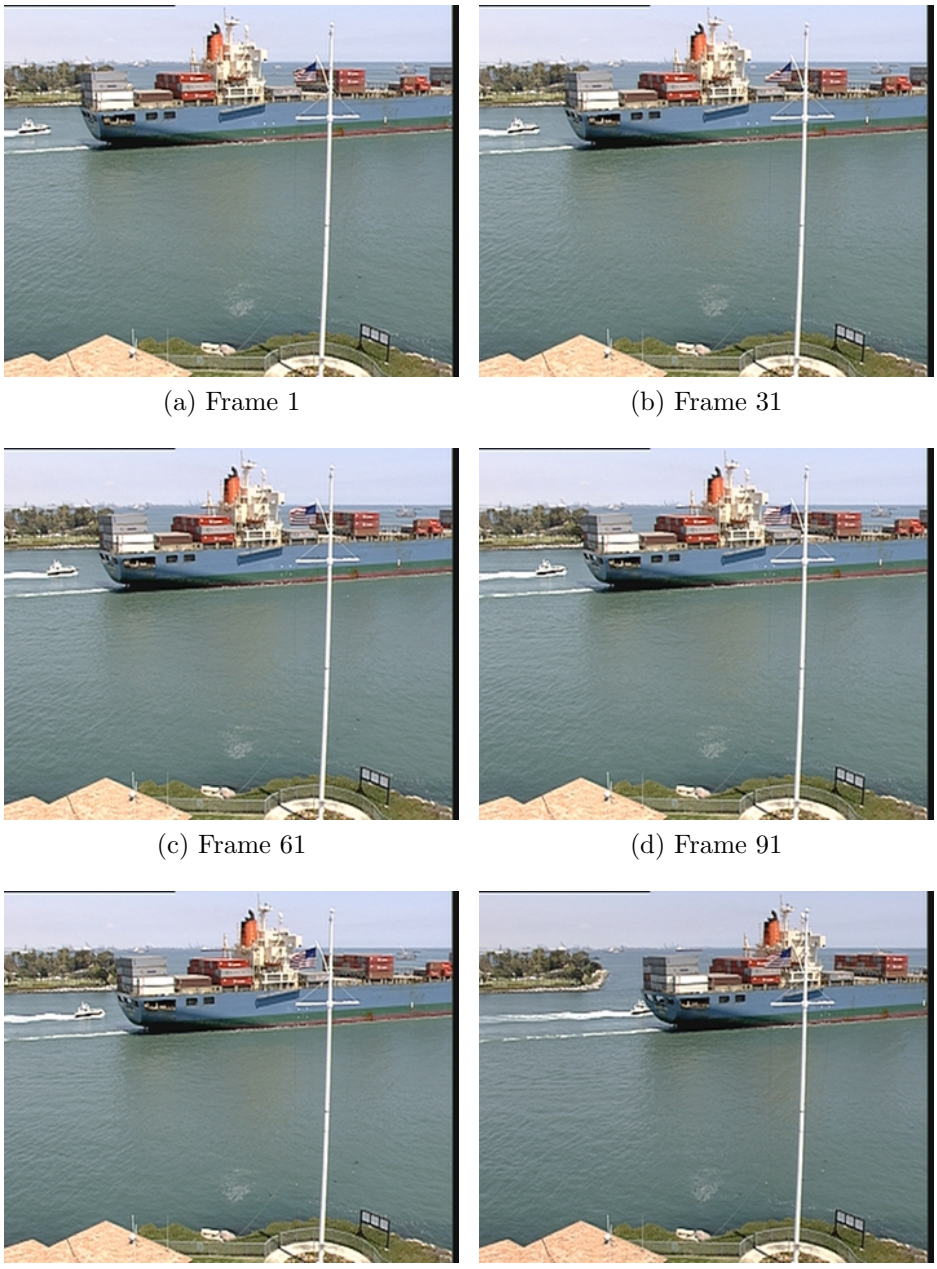
All sequences are in CIF resolution at 30 Hz, they are 10 seconds long. Six frames of each video sequence are depicted in the sequel. Although the color versions are shown bellow, only the luma component is coded.

(a) Frame 1 (b) Frame 31

(c) Frame 61 (d) Frame 91

(e) Frame 121 (f) Frame 300

Figure B.1: Frames from the video sequence: *Akiyo*



(a) Frame 1 (b) Frame 31

(c) Frame 61 (d) Frame 91

(e) Frame 121 (f) Frame 300

Figure B.2: Frames from the video sequence: *Container*

(a) Frame 1

(b) Frame 31

(c) Frame 61

(d) Frame 91

(e) Frame 121

(f) Frame 300

Figure B.3: Frames from the video sequence: *Hall*

(a) Frame 1 (b) Frame 31

(c) Frame 61 (d) Frame 91

(e) Frame 121 (f) Frame 300

Figure B.4: Frames from the video sequence: *Paris*

Appendix C

Seamless image stitching with Poisson equation

This Appendix briefly describes the seamless patch stitching technique used in the this thesis. The objective is to keep the text somewhat more self-contained. Refer to [109] for more in-depth treatment.

C.1 Problem statement

Let t, the scalar target function that one wants to interpolate, be defined over S minus the interior of Ω (for short $S\backslash\Omega$); let $\mathbf{v}$ be a guidance vector field defined over the interior of Ω and f the unknown interpolating function defined over the interior Ω. Only the values of f over the domain Ω must be determined as the target function t should be kept unchanged over $S\backslash\Omega$. Figure C.1 summarizes the definitions given above.

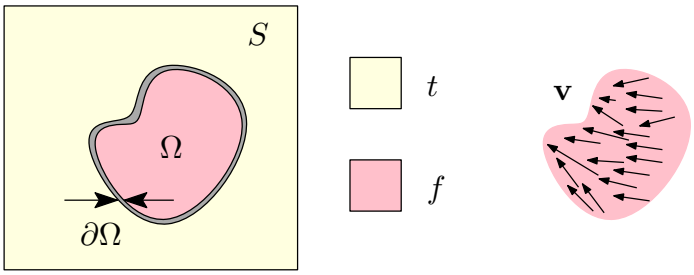


Figure C.1: Poisson editing: problem statement.

The objective is to find a function f whose gradient is as close as possible to the guidance vector field $\mathbf{v}$ subject to the condition that f equals t in the boundary $\partial\Omega$ of Ω.

The mathematical formulation for this problem is described in [109, 120, 121] as follows:

$$\min_f \int_\Omega |\nabla f - \mathbf{v}|^2, \text{ with } f|_{\partial\Omega} = t|_{\partial\Omega} \tag{C.1}$$

The above solution is also the solution to the Poisson partial differential equation with Dirichlet boundary conditions:

$$\Delta f = \text{div}\mathbf{v}, \text{ with } f|_{\partial\Omega} = t|_{\partial\Omega} \tag{C.2}$$

where $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ is Laplacian of f and div$\mathbf{v}$ is the divergence of the guidance vector field.

In the usual case for seamless stitching, the guidance field is chosen to be the gradient of the source function g, then:

$$\Delta f = \Delta g, \text{ with } f|_{\partial\Omega} = t|_{\partial\Omega} \tag{C.3}$$

It turns out that the discrete solution for such problem amounts to solve the linear system:

$$|N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} t_q + \sum_{q \in N_p} v_{pq}, \text{ with } q \neq p \tag{C.4}$$

where p is a sample position in S, N_p the set of 4-connected neighbors to p, q is a sample position in N_p, $|N_p|$ the number of available samples in the 4-connected neighbors set and v_{pq} is the gradient of the source image g approximated by $v_{pq} = g_p - g_q$.

Let us consider the working example schematically depicted in Figure C.2

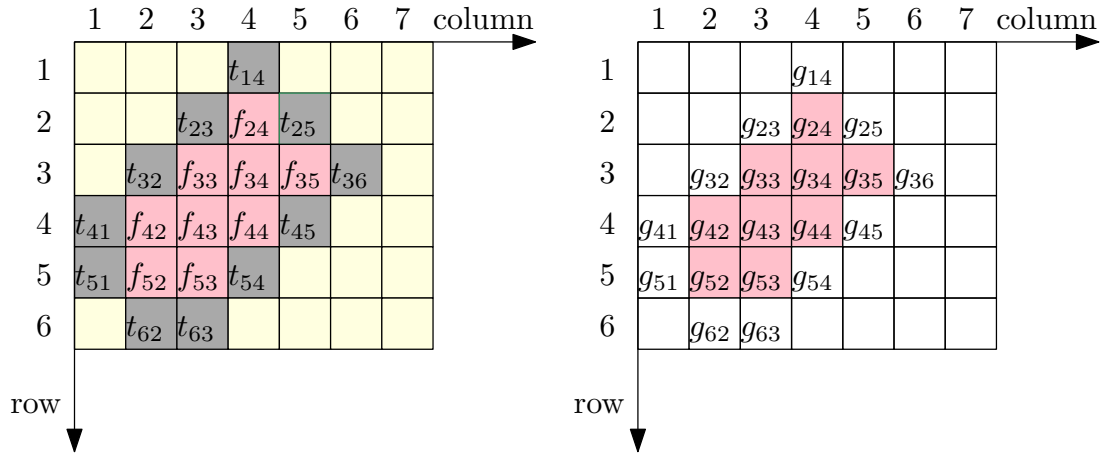


Figure C.2: Discrete seamless image stitching.

Considering Equation C.4, let us write down the equations for a few samples within Ω.

For f_{24}, it follows:

$$4f_{24} - f_{34} = t_{14} + t_{23} + t_{25} + [(g_{24} - g_{14}) + (g_{24} - g_{23}) + (g_{24} - g_{25}) + (g_{24} - g_{34})]$$
$$4f_{24} - f_{34} = t_{14} + t_{23} + t_{25} + (4g_{24} - g_{14} - g_{23} - g_{25} - g_{34}) \quad \text{(C.5)}$$

For f_{33}, it follows:

$$4f_{33} - f_{34} - f_{43} = t_{23} + t_{32} + (4g_{33} - g_{23} - g_{32} - g_{34} - g_{43}) \quad \text{(C.6)}$$

For f_{34}, it follows:

$$4f_{34} - f_{24} - f_{33} - f_{35} - f_{44} = (4g_{34} - g_{24} - g_{33} - g_{35} - g_{44}) \quad \text{(C.7)}$$

Continuing on this, one gets:

$$A\mathbf{x} = \mathbf{b} \quad \text{(C.8)}$$

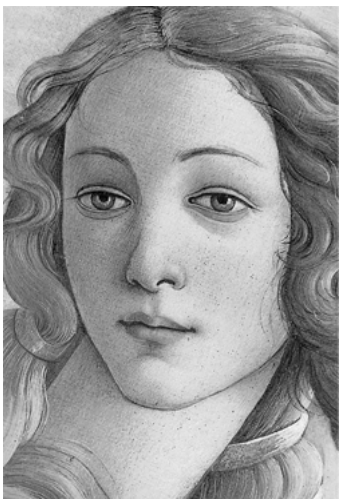
where $A = \begin{bmatrix} 4 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}$; $\mathbf{x} = \begin{bmatrix} f_{24} \\ f_{33} \\ f_{34} \\ f_{35} \\ f_{42} \\ f_{43} \\ f_{44} \\ f_{52} \\ f_{53} \end{bmatrix}$

and $\mathbf{b} = \begin{bmatrix} t_{14} + t_{23} + t_{25} + (4g_{24} - g_{14} - g_{23} - g_{25} - g_{34}) \\ t_{23} + t_{32} + (4g_{33} - g_{23} - g_{32} - g_{34} - g_{43}) \\ 4g_{34} - g_{24} - g_{33} - g_{35} - g_{44} \\ t_{25} + t_{36} + t_{45} + (4g_{35} - g_{25} - g_{34} - g_{36} - g_{45}) \\ t_{32} + t_{41} + (4g_{42} - g_{32} - g_{41} - g_{43} - g_{52}) \\ 4g_{43} - g_{33} - g_{42} - g_{44} - g_{53} \\ t_{45} + t_{54} + (4g_{44} - g_{34} - g_{43} - g_{45} - g_{54}) \\ t_{51} + t_{62} + (4g_{52} - g_{42} - g_{51} - g_{53} - g_{62}) \\ t_{54} + t_{63} + (4g_{53} - g_{43} - g_{52} - g_{54} - g_{63}) \end{bmatrix}$

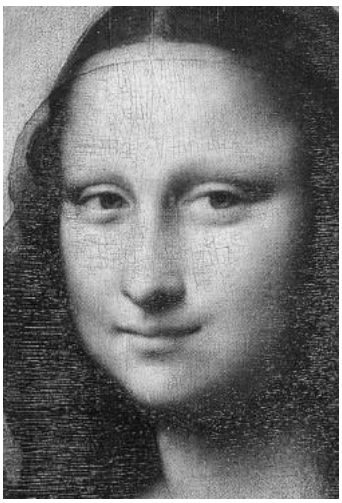
The matrix A is symmetric and positive-definite [109], therefore the conjugate gradient method [110, 112, 113] has been adopted for solving the linear system $A\mathbf{x} = \mathbf{b}$.

Regarding the time-complexity, let us consider the example depicted in Fig-

ure C.3. In this example, the number of luma samples to be stitched is 29928. The conjugate gradient method reaches the solution in 0.15 seconds. Figure C.3 shows the results for 10, 50 and 100 iterations. One can notice that after 50 iterations the seam has already disappeared.



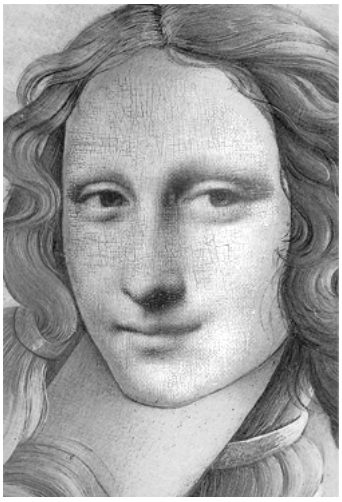
(a) Source image.



(b) Destination image.

(c) 10 iterations.

(d) 50 iterations.



(e) 100 iterations.

Figure C.3: Discrete seamless image stitching.

Appendix D

Additional results

D.1 Trading-off visualization and searching performances

Figures D.1, D.2, D.3 and D.4 show additional results to the ones presented in Section 5.4. One may notice that these results also confirm the remarks discussed in Section 5.4.

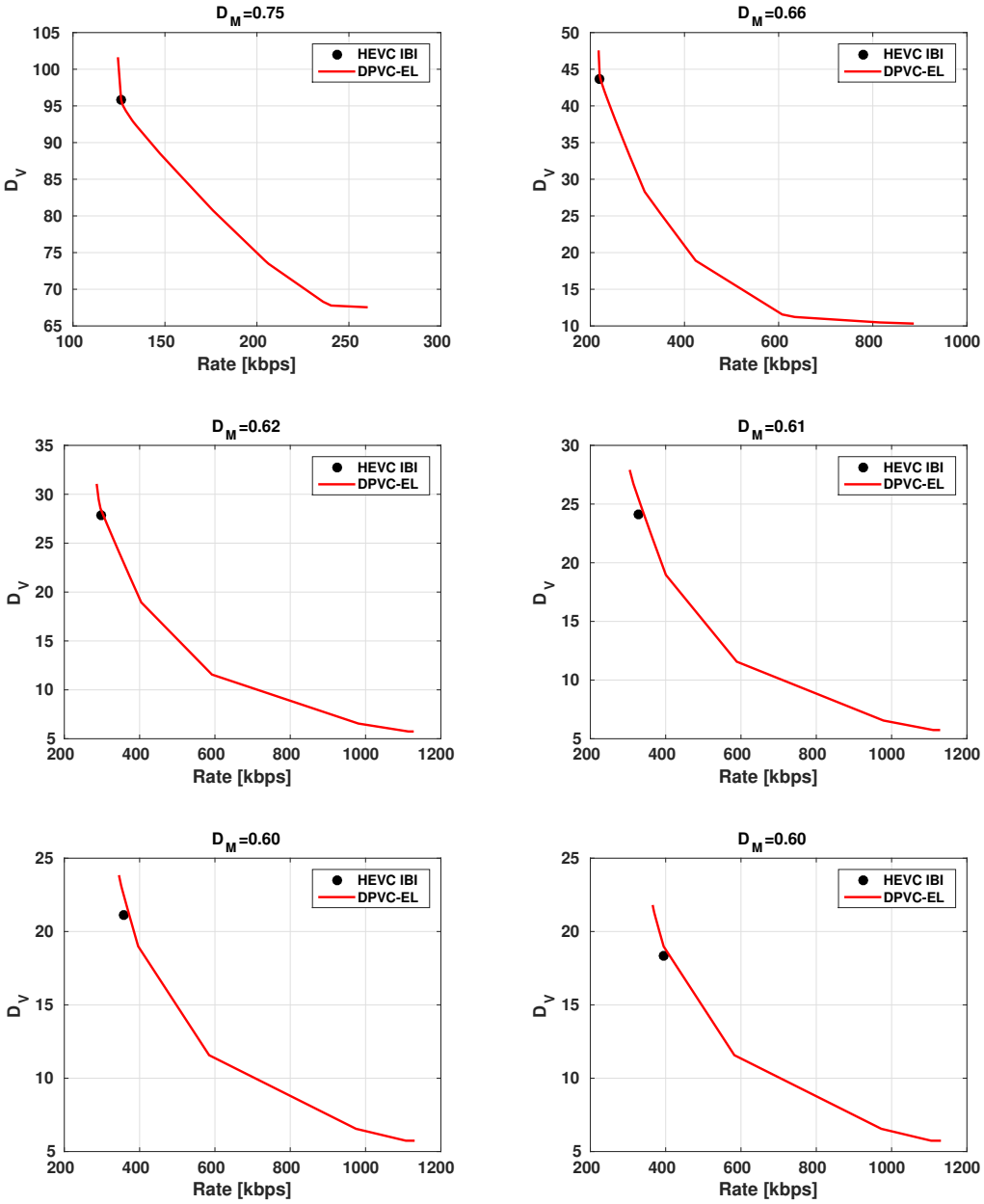


Figure D.1: RD_V performance for fixed descriptor matching performances D_M: sequence *Hall*.

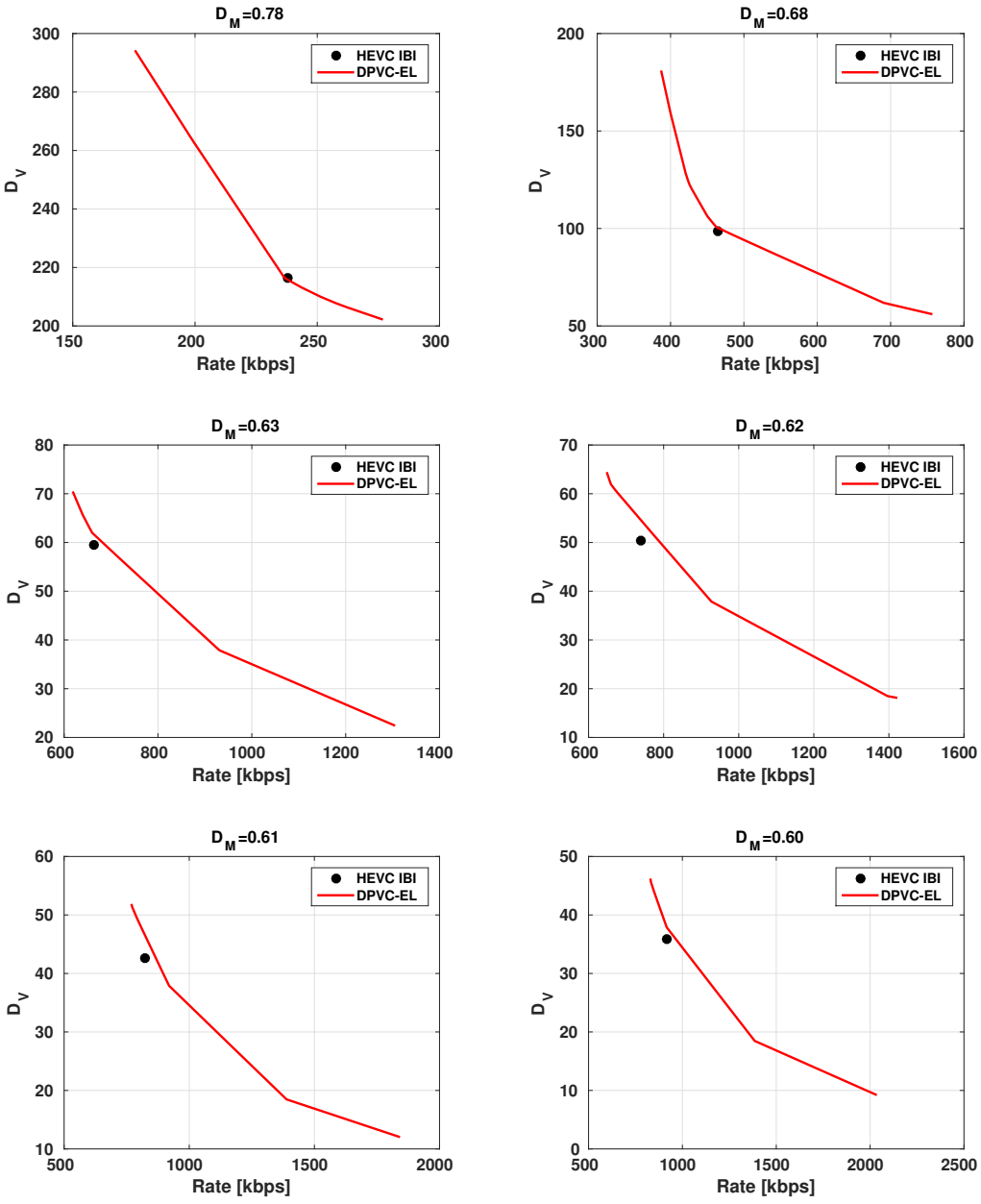


Figure D.2: RD_V performance for fixed descriptor matching performances D_M: sequence *Paris*.

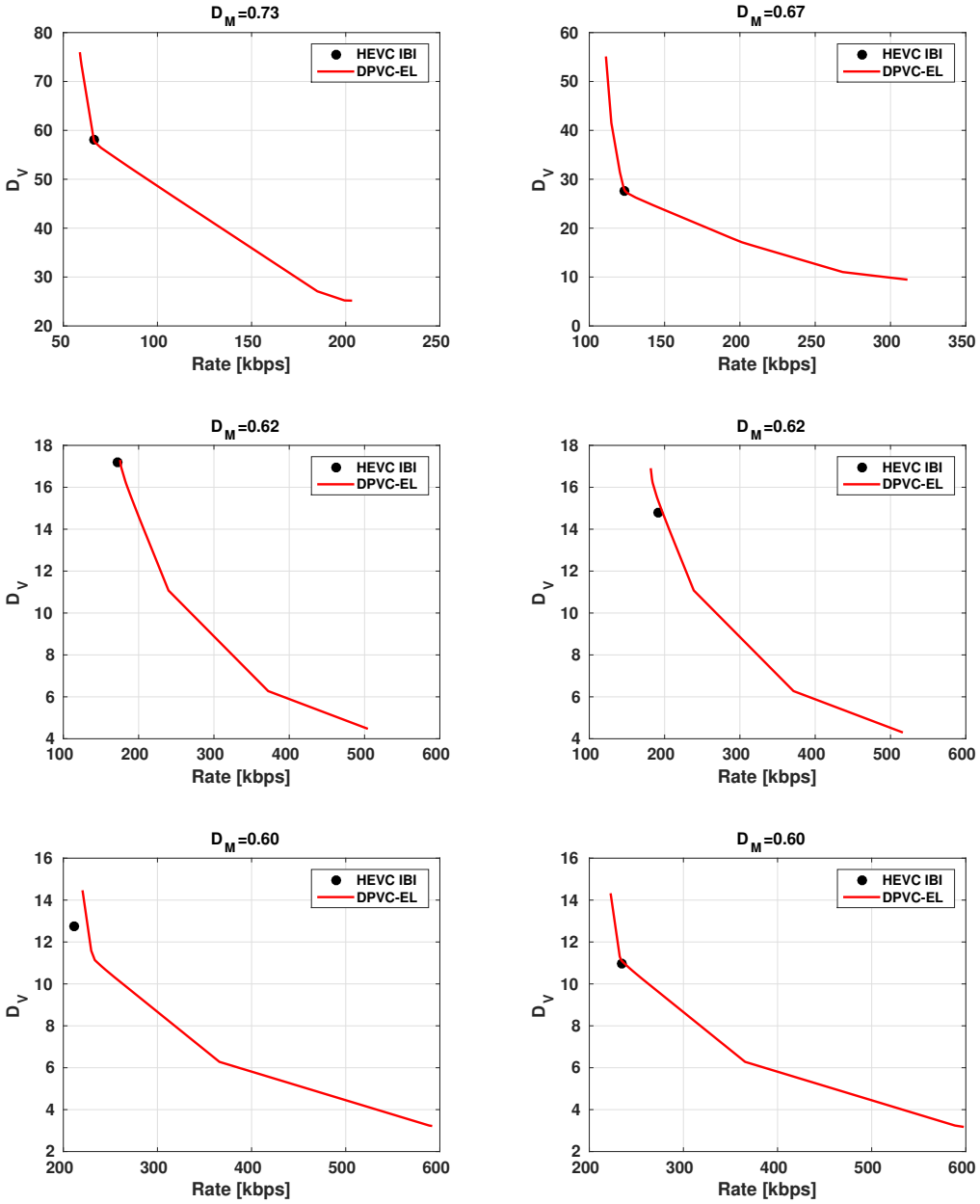


Figure D.3: RD_V performance for fixed descriptor matching performances D_M: sequence *Akiyo*.

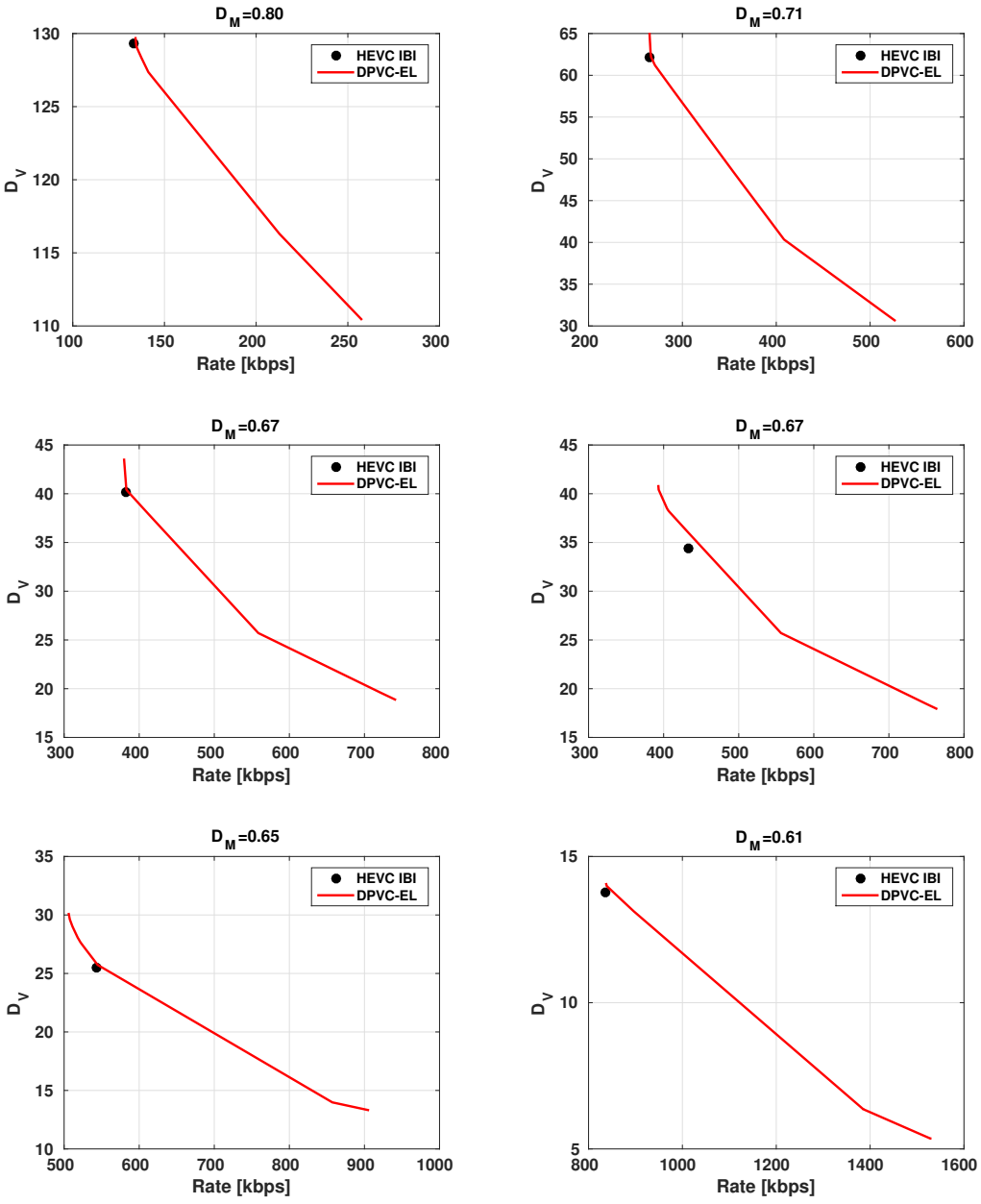
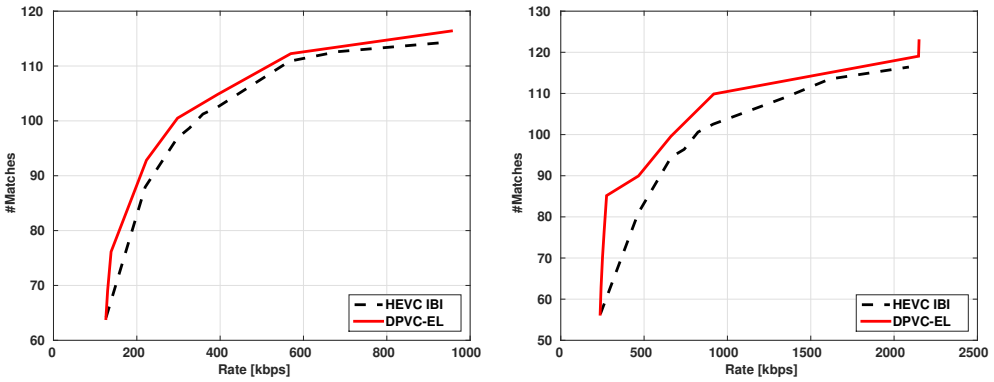


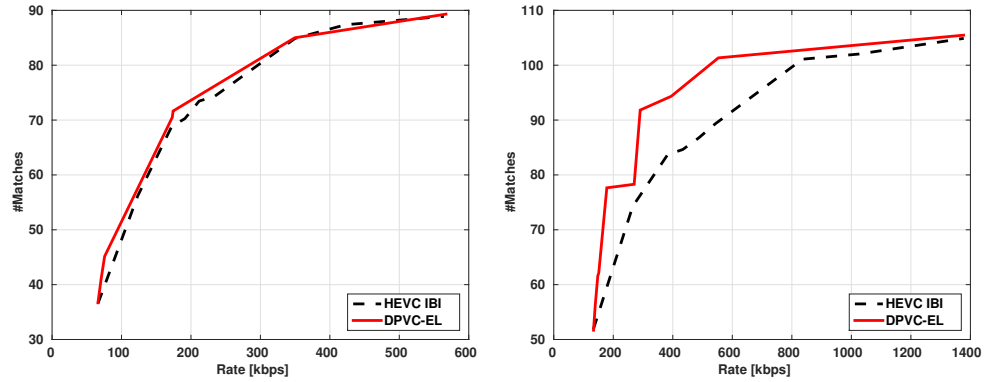
Figure D.4: RD_V performance for fixed descriptor matching performances D_M: sequence *Container*.

D.2 Best searching performance

Figure D.5 shows the number of matches as function of the bitrate for the operational points selected to provide the best RD_M performance as presented in Section 5.5. Whereas Figures D.6, D.7, D.8, D.9, D.10, D.11 and D.12 depict feature matches between sample frames from the used sequences and their reference image in the database.

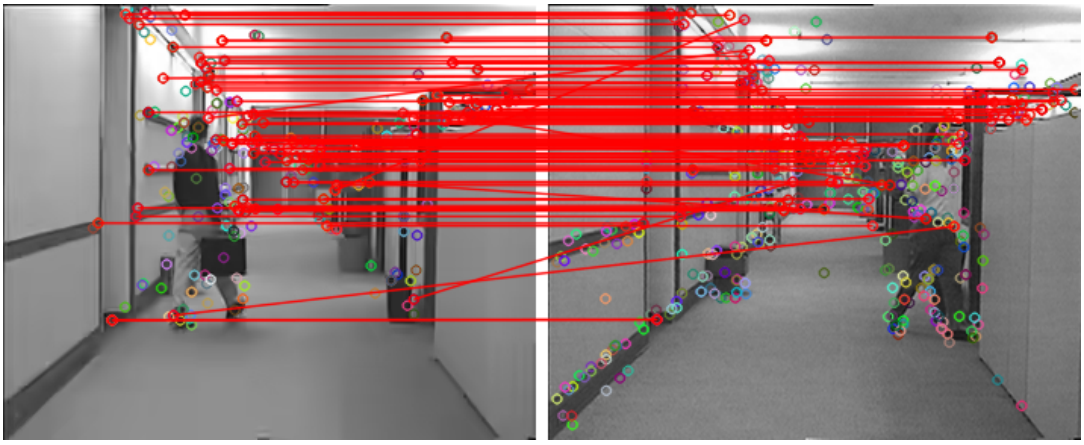


(a) *Hall* (b) *Paris*

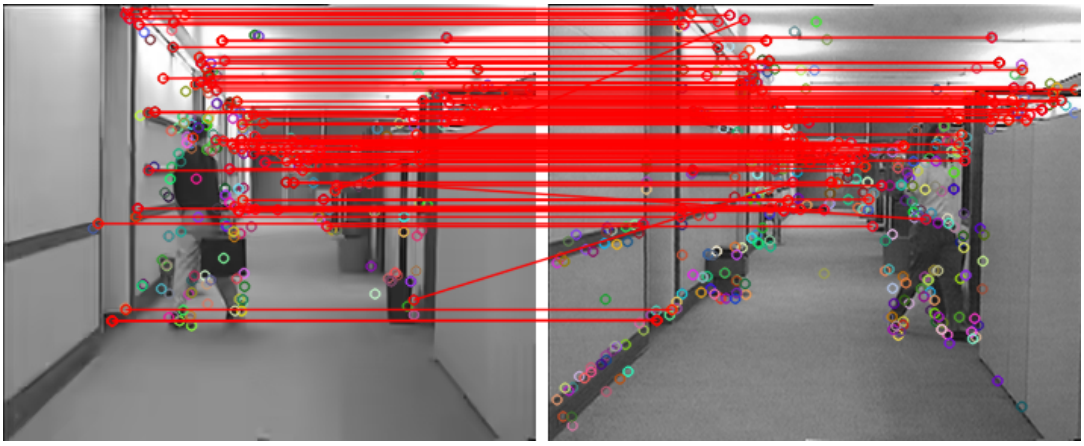


(c) *Akiyo* (d) *Container*

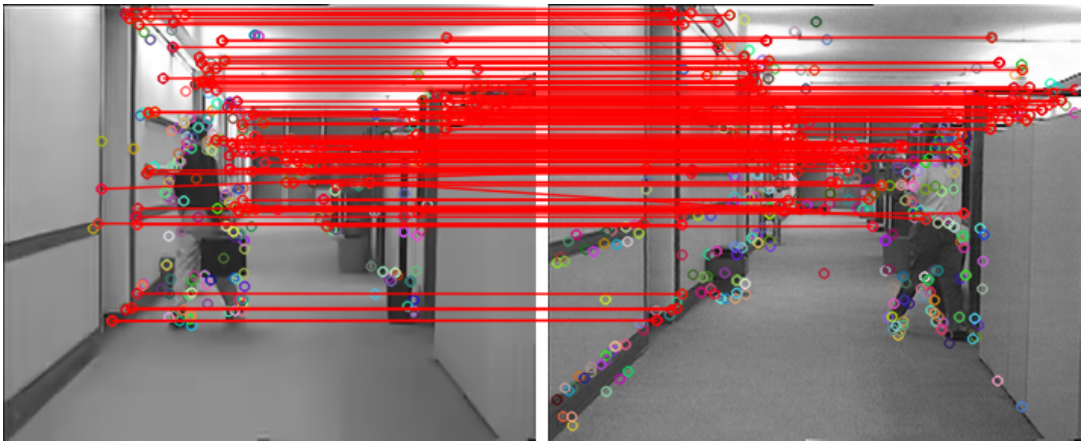
Figure D.5: Best operational points in terms of Rate-#Matches performance obtained from the convex hull points RD_M for sequences *Hall*, *Paris*, *Akiyo* and *Container*.



(a) Frame 43, QP = 37, $\lambda = 2^{-5}$ and $\gamma = 50$

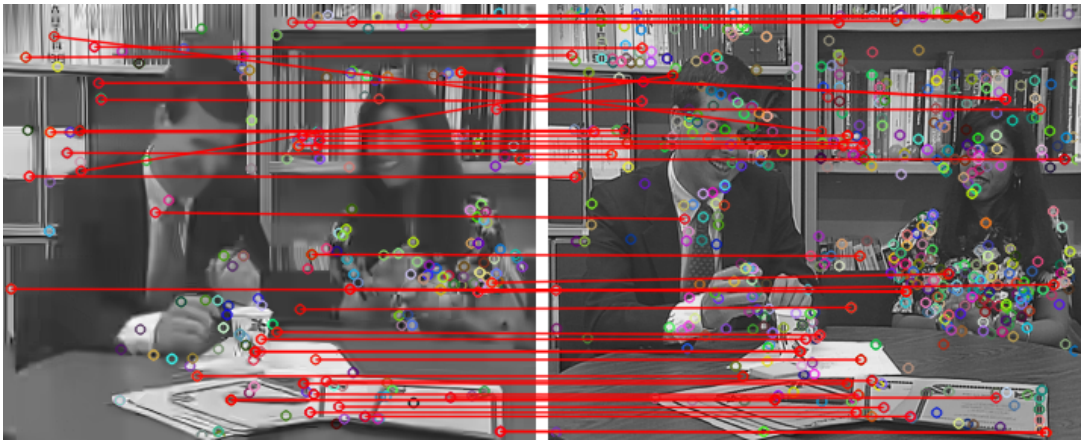


(b) Frame 43, QP = 37, $\lambda = 2^{-10}$ and $\gamma = 50$

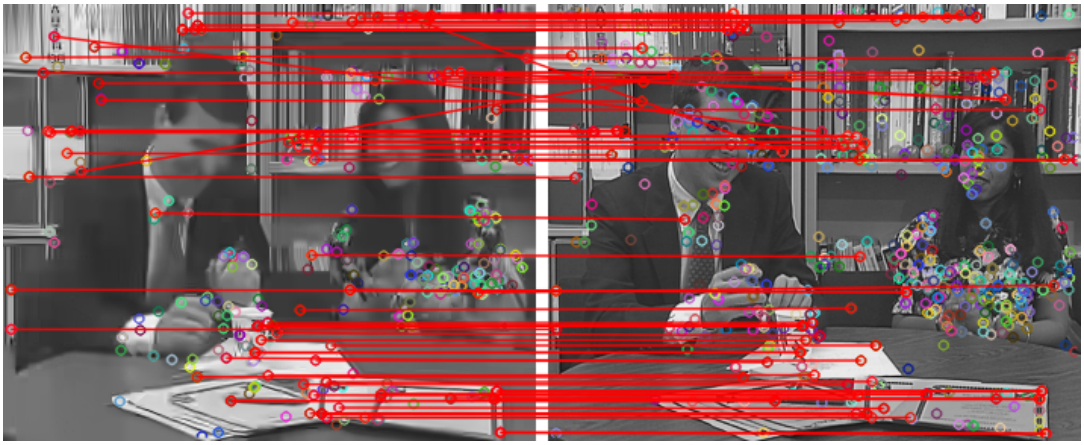


(c) Frame 43, QP = 37, $\lambda = 2^{-20}$ and $\gamma = 50$

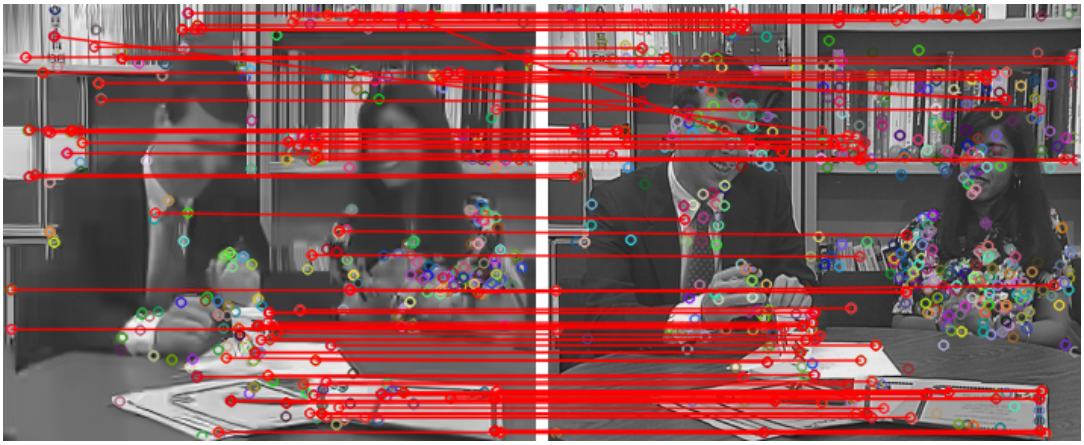
Figure D.6: Feature matches between a frame of the sequence *Hall* and its reference image in the database, k-frames coded with QP=37.



(a) Frame 63, QP = 45, $\lambda = 2^{-5}$ and $\gamma = 50$



(b) Frame 63, QP = 45, $\lambda = 2^{-10}$ and $\gamma = 50$

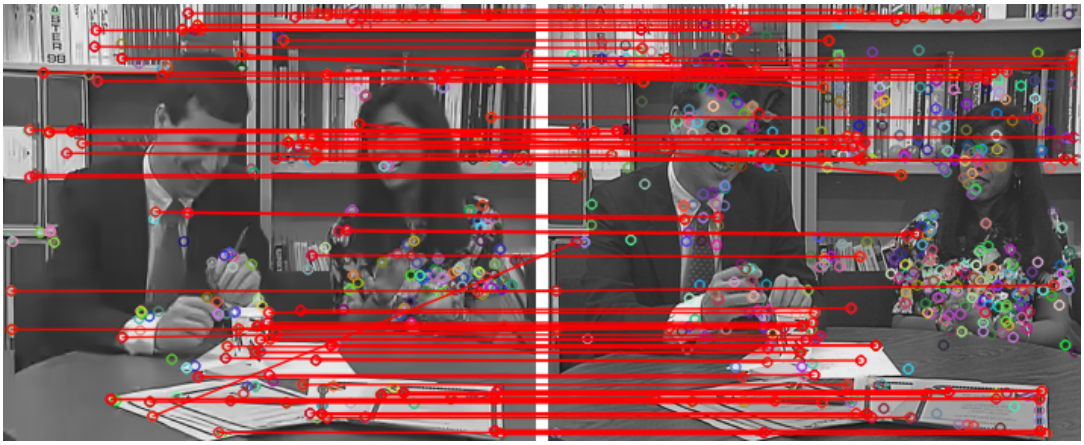


(c) Frame 63, QP = 45, $\lambda = 2^{-20}$ and $\gamma = 50$

Figure D.7: Feature matches between a frame of the sequence *Paris* and its reference image in the database, k-frames coded with QP=45.

(a) Frame 63, QP = 37, $\lambda = 2^{-5}$ and $\gamma = 50$

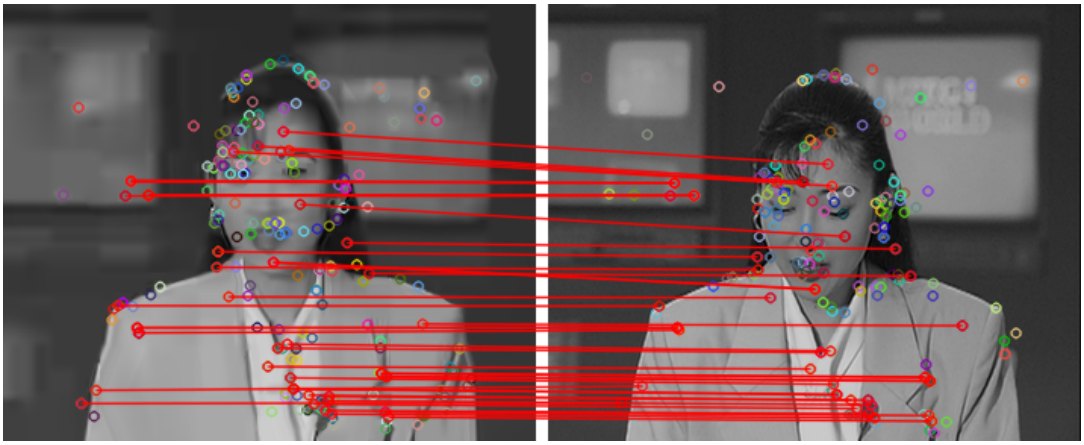
(b) Frame 63, QP = 37, $\lambda = 2^{-10}$ and $\gamma = 50$



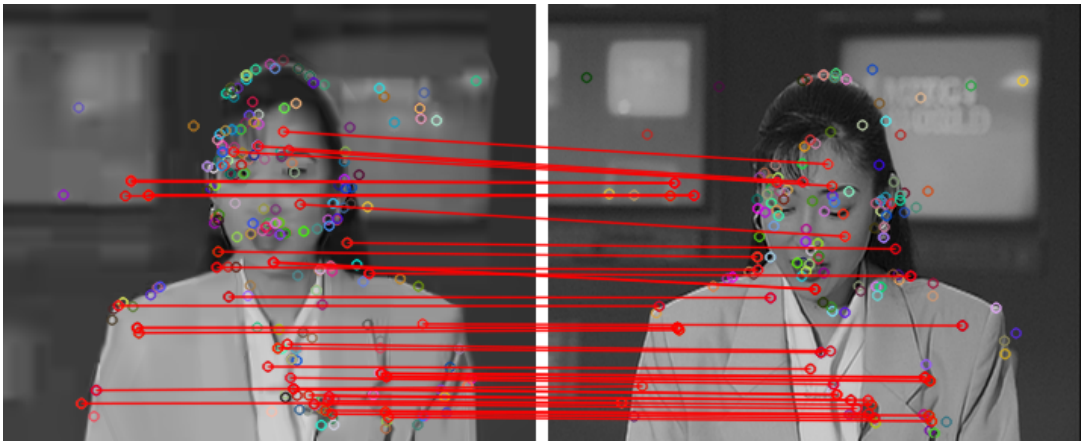
(c) Frame 63, QP = 37, $\lambda = 2^{-20}$ and $\gamma = 50$

Figure D.8: Feature matches between a frame of the sequence *Paris* and its reference image in the database, k-frames coded with QP=37.

(a) Frame 21, QP = 45, $\lambda = 2^{-5}$ and $\gamma = 50$

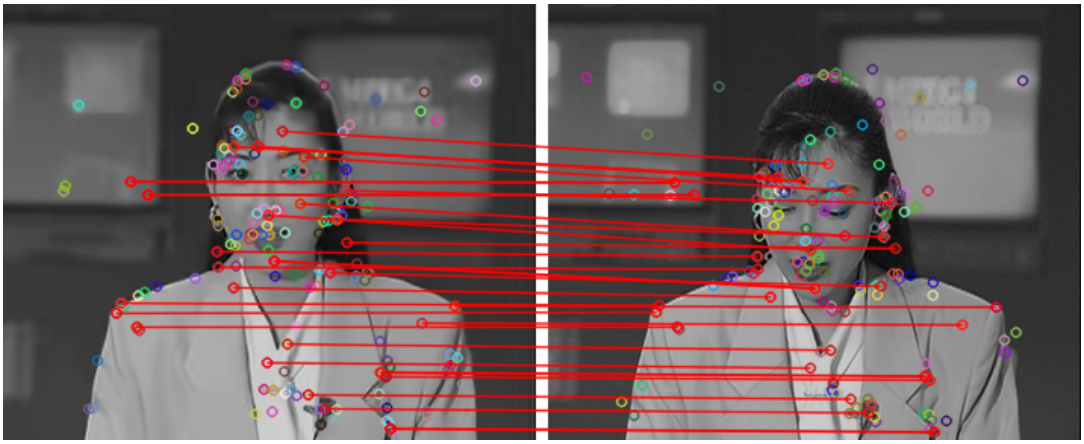


(b) Frame 21, QP = 45, $\lambda = 2^{-10}$ and $\gamma = 50$

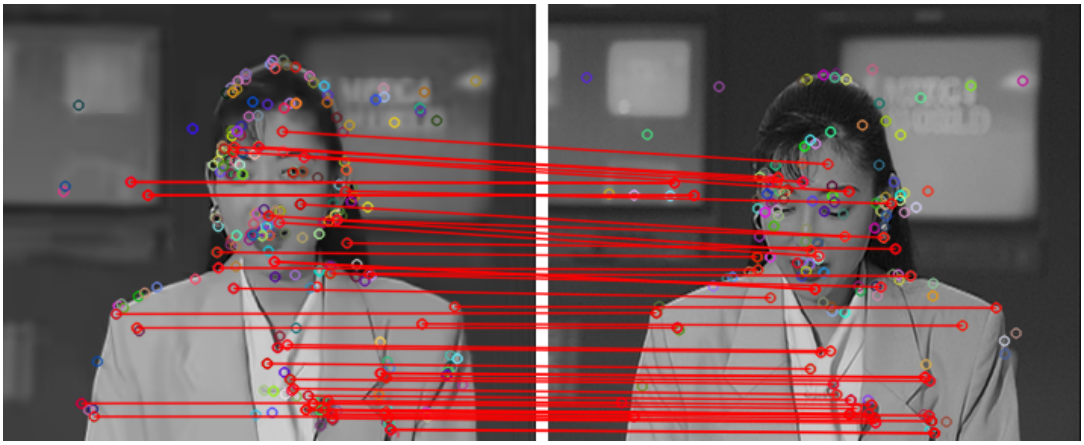


(c) Frame 21, QP = 45, $\lambda = 2^{-20}$ and $\gamma = 50$

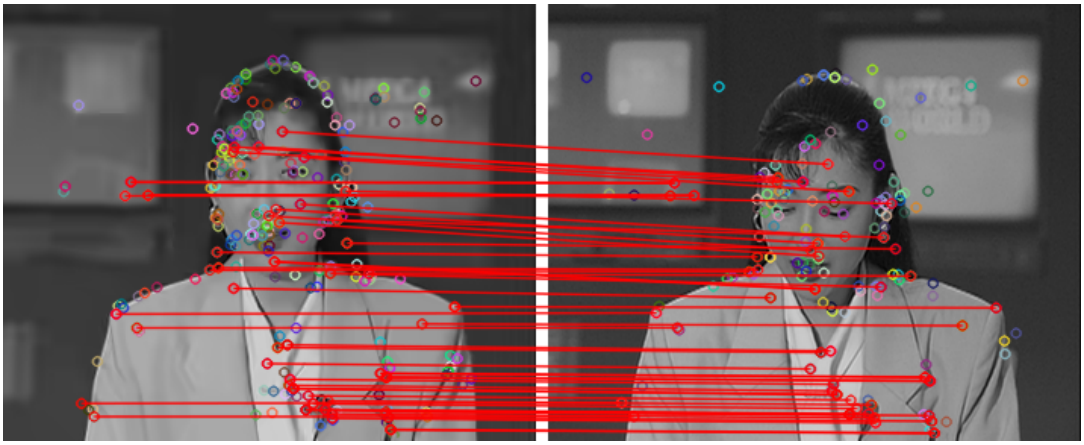
Figure D.9: Feature matches between a frame of the sequence *Akiyo* and its reference image in the database, k-frames coded with QP=45.



(a) Frame 21, QP = 37, $\lambda = 2^{-5}$ and $\gamma = 50$

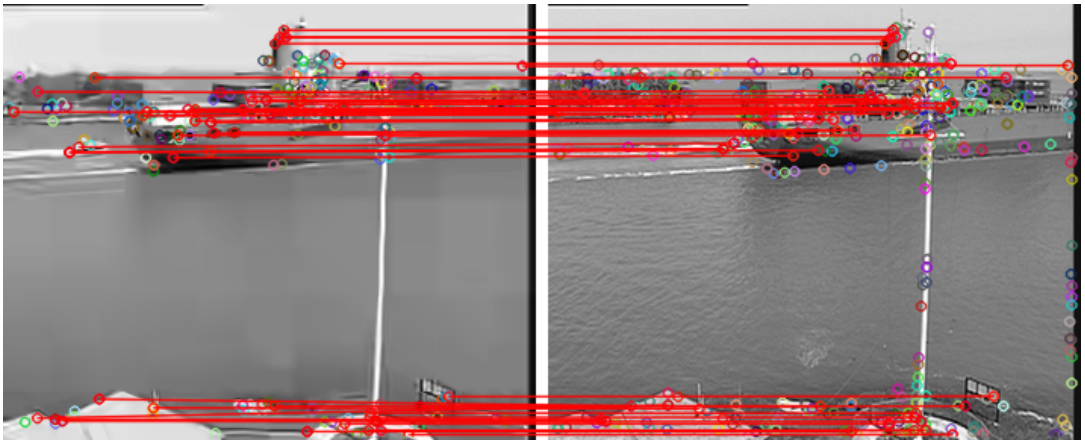


(b) Frame 21, QP = 37, $\lambda = 2^{-10}$ and $\gamma = 50$

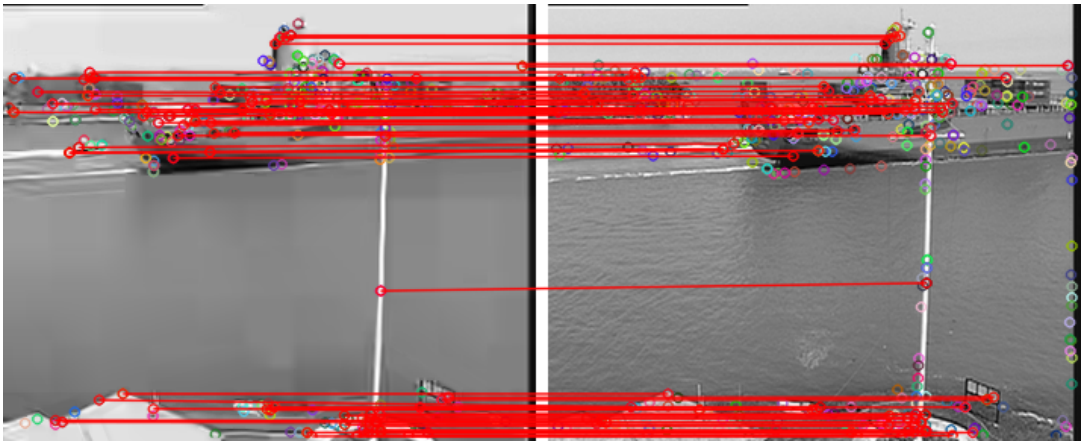


(c) Frame 21, QP = 37, $\lambda = 2^{-20}$ and $\gamma = 50$

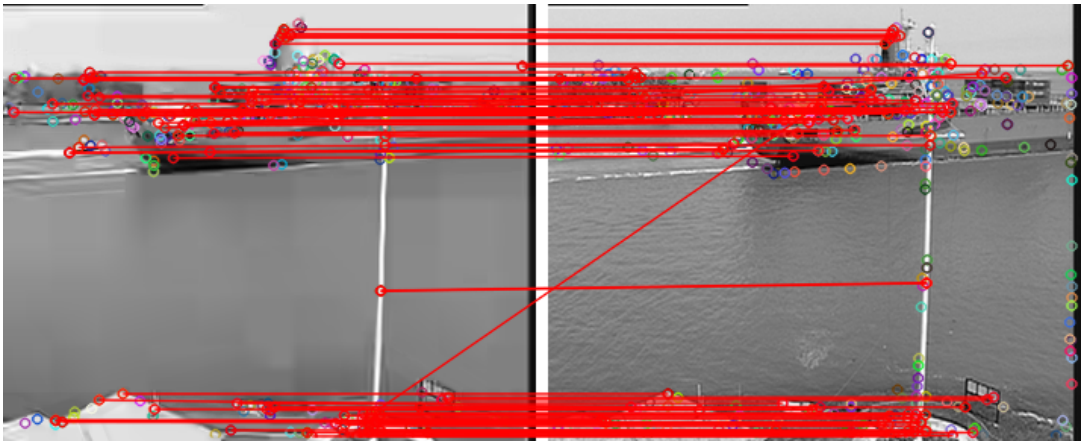
Figure D.10: Feature matches between a frame of the sequence *Akiyo* and its reference image in the database, k-frames coded with QP=37.



(a) Frame 95, QP = 45, $\lambda = 2^{-5}$ and $\gamma = 50$

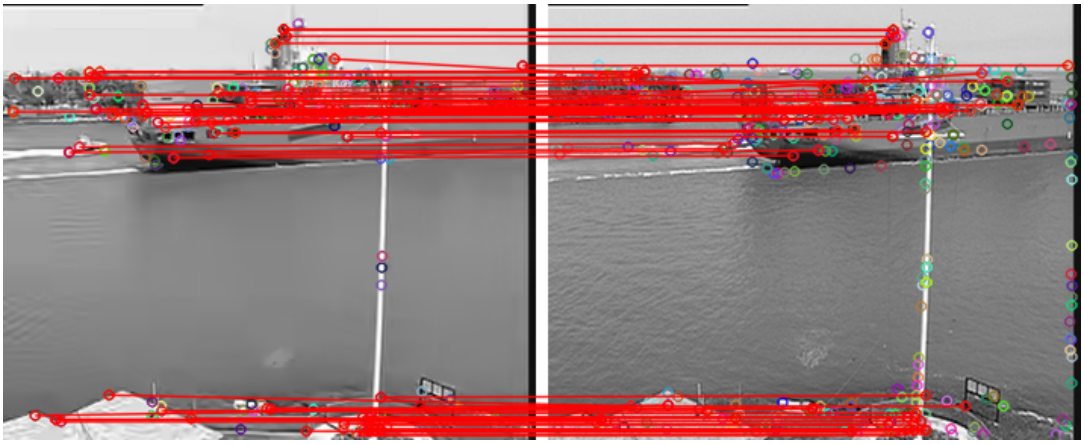


(b) Frame 95, QP = 45, $\lambda = 2^{-10}$ and $\gamma = 50$

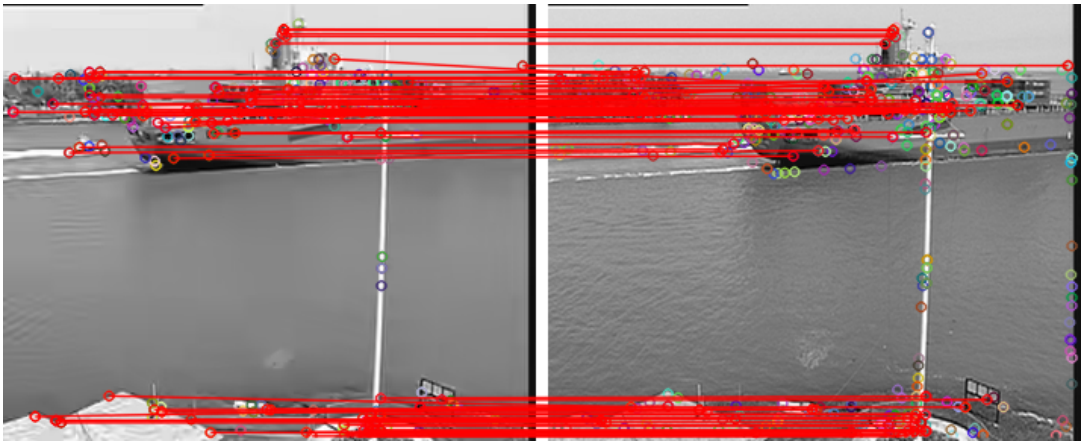


(c) Frame 95, QP = 45, $\lambda = 2^{-20}$ and $\gamma = 50$

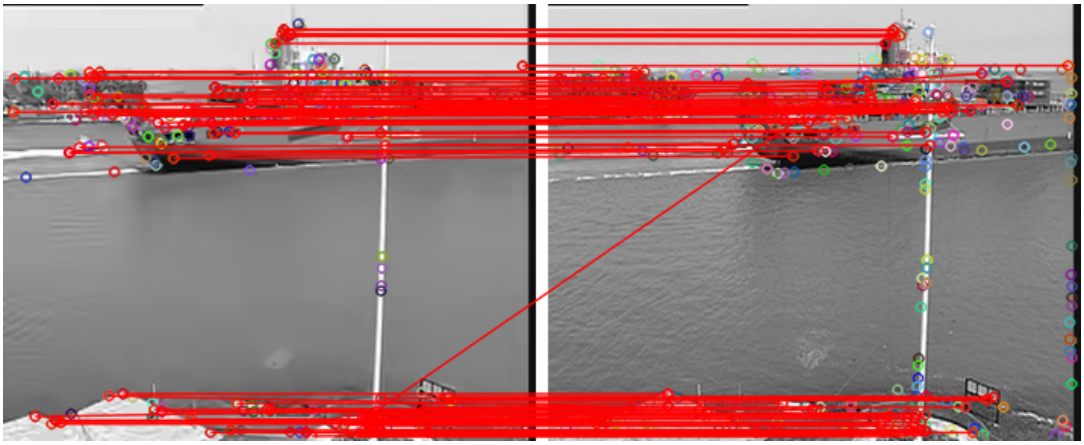
Figure D.11: Feature matches between a frame of the sequence *Container* and its reference image in the database, k-frames coded with QP=45.



(a) Frame 95, QP = 37, $\lambda = 2^{-5}$ and $\gamma = 50$



(b) Frame 95, QP = 37, $\lambda = 2^{-10}$ and $\gamma = 50$



(c) Frame 95, QP = 37, $\lambda = 2^{-20}$ and $\gamma = 50$

Figure D.12: Feature matches between a frame of the sequence *Container* and its reference image in the database, k-frames coded with QP=37.

Bibliography

[1] ORTEGA, A., RAMCHANDRAN, K. "Rate-Distortion Methods for Image and Video Compression", *IEEE Signal Processing Magazine*, v. 15, n. 6, pp. 23–50, November 1988.

[2] SULLIVAN, G. J., WIEGAND, T. "Rate-Distortion Optimization for Video Compression", *IEEE Signal Processing Magazine*, v. 15, n. 6, pp. 74–90, November 1998.

[3] SULLIVAN, G. J., OHM, J.-R., HAN, W.-J., et al. "Overview of the High Efficiency Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1649–1668, December 2012.

[4] BOSSE, F., BROSS, B., SUHRING, K., et al. "HEVC Complexity and Implementation Analysis", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1685–1696, December 2012.

[5] RICHARDSON, I. E. G. *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. 1 ed. West Sussex, England, John Wiley and Sons Ltd., 2003.

[6] GIROD, B., CHANDRASEKHAR, V., CHEN, D. M., et al. "Mobile Visual Search: Linking the Virtual and Physical Worlds", *IEEE Signal Processing Magazine*, v. 28, n. 4, pp. 61–76, July 2011.

[7] LOWE, D. G. "Object Recognition from Local Scale-Invariant Features". In: *International Conference on Computer Vision*, Kerkyra, Greece, September 1999.

[8] LOWE, D. G. "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, v. 60, n. 2, pp. 91–110, November 2004.

[9] CHEN, S. D., MOULIN, P. "A Two-part Predictive Coder for Multitask Signal Compression". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014.

[10] BAY, H., TUYTELAARS, T., GOOL, L. V. "SURF: Speeded Up Robust Features". In: *9th European Conference on Computer Vision*, Graz, Austria, June 2008.

[11] BAY, H., ESS, A., TUYTELAARS, T., et al. "Speeded-Up Robust Features", *Journal Computer Vision and Image Understanding*, v. 110, n. 3, pp. 346–359, June 2008.

[12] "Information Technology – Multimedia Content Description Interface – Part 13: Compact Descriptors for Visual Search". ISO/IEC JTC 1/SC 29 N 13925. 2013-12-13.

[13] DUAN, L.-Y., CHANDRASEKHAR, V., CHEN, J., et al. "Overview of the MPEG-CDVS Standard", *IEEE Transactions on Image Processing*, v. 25, n. 1, pp. 179–194, November 2015.

[14] BAROFFIO, L., CESANA, M., REDONDI, A., et al. "Coding Visual Features Extracted from Video Sequences", *IEEE Transactions on Image Processing*, v. 23, n. 5, pp. 2262–2276, March 2014.

[15] CHAO, J., STEINBACH, E. "Keypoint Encoding and Transmission for Improved Feature Extraction from Compressed Images". In: *IEEE International Conference on Multimedia and Expo (ICME)*, Turin, Italy, August 2015.

[16] BAROFFIO, L., CESANA, M., REDONDI, A., et al. "Hybrid Coding of Visual Content and Local Image Features". In: *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, September 2015.

[17] CHAO, J., STEINBACH, E. "Preserving SIFT Features in JPEG-encoded Images". In: *International Conference on Image Processing (ICIP)*, Brussels, Belgium, September 2011.

[18] MAKAR, M., LAKSHMAN, H., CHANDRASEKHAR, V., et al. "Gradient Preserving Quantization". In: *International Conference on Image Processing (ICIP)*, Orlando, USA, September 2012.

[19] CHAO, J., STEINBACH, E. "SIFT Feature-preserving Bit Allocation for H.264/AVC Video Compression". In: *International Conference on Image Processing (ICIP)*, Orlando, USA, September 2012.

[20] ZHANG, X., MA, S., WANG, S., et al. "A Joint Compression Scheme of Video Feature Descriptors and Visual Content", *IEEE Transactions on Image Processing*, v. 26, n. 2, pp. 633–647, February 2017.

[21] REDONDI, A., CESANA, M., TAGLIASACCHI, M. "Low Bitrate Coding Schemes for Local Image Descriptors". In: *IEEE 14th International Workshop on Multimedia Signal Processing*, Banff, Canada, September 2012.

[22] CHANDRASEKHAR, V., TAKACS, G., CHEN, D., et al. "Transform Coding of Image Feature Descriptors". In: *Proc. SPIE 7257*, January 2009.

[23] CHANDRASEKHAR, V., TAKACS, G., CHEN, D. M., et al. "Compressed Histogram of Gradients: A Low-Bitrate Descriptor", *International Journal of Computer Vision*, v. 96, n. 3, pp. 384–399, February 2012.

[24] ASCENSO, J., PEREIRA, F. "Lossless Compression of Binary Image Descriptors for Visual Sensor Networks". In: *18th International Conference on Digital Signal Processing*, Fira, Greece, July 2013.

[25] YUE, H., SUN, X., WU, F., et al. "SIFT-based Image Compression". In: *IEEE International Conference on Multimedia and Expo*, Melbourne, Australia, July 2012.

[26] YUE, H., SUN, X., YANG, J., et al. "Cloud-Based Image Coding for Mobile Devices – Toward Thousands to One Compression", *IEEE Transactions on Multimedia*, v. 15, n. 4, pp. 845–857, June 2013.

[27] WEINZAEPFEL, P., JÉGOU, H., PÉREZ, P. "Reconstructing an Image from its Local Descriptors". In: *2011 IEEE Conference on Computer Vision and Pattern Recoginition*, Colorado Springs, USA, June 2011.

[28] D'ANGELO, E., JACQUES, L., ALAHI, A., et al. "From Bits to Images: Inversion of Local Binary Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 5, pp. 874–887, May 2014.

[29] MILANI, S., AGRESTI, G., CALVAGNO, G. "A Rate Control Algorithm for Video Coding in Augmented Reality Applications". In: *Picture Coding Symposium (PCS)*, Nuremberg, Germany, December 2016.

[30] CHAO, J., ECKEHARD. "Keypoint Encoding for Improved Feature Extraction From Compressed Video at Low Bitrates", *IEEE Transactions on Multimedia*, v. 18, n. 1, pp. 25–39, January 2016.

[31] CHAO, J. *Feature-preserving Image and Video Compression*. Phd thesis, Technical University of Munich, Munich, Germany, 2015.

[32] "Displays Of A Different Stripe: New Image-Management Techniques Provide Brightness for Half the Power, Easing the Brutal Trade-off in Video-Rich Gadgets". `https://spectrum.ieee.org/computing/hardware/displays-of-a-different-stripe`. Accessed in September, 2017.

[33] BLINN, J. "What is a Pixel?" *IEEE Computer Graphics and Applications*, v. 25, n. 5, pp. 82–87, September 2005.

[34] HEALEY, G. E., KONDEPUDY, R. "Radiometric CCD Camera Calibration and Noise Estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 3, pp. 267–276, March 1994.

[35] SZELISKI, R. *Computer Vision: Algorithms and Applications*. 1 ed. Washington, USA, Springer-Verlag, 2011.

[36] ITU-R. *Recommendation ITU-R BT.601-7*. ITU-T Radiocommunication Sector of ITU, March 2011.

[37] COVER, T. M., THOMAS, J. A. *Elements of Information Theory*. 2 ed. New Jersey, USA, John Wiley and Sons, 2006.

[38] SHANNON, C. E. "A Mathematical Theory of Communication", *The Bell System Technical Journal*, v. 27, pp. 379–423, July 1948.

[39] SHANNON, C. E. "Coding Theorems for a Discrete Source with a Fidelity Criterion", *IRE International Convention Record*, v. 7, pp. 142–163, 1959.

[40] KHALID SAYOOD. *Introduction to Data Compression*. 3 ed. San Francisco, USA, Elsevier Inc., 2006.

[41] ITU-T. *Advanced Video Coding for Generic Audiovisual Services*. Recommendation ITU-T H.264, March 2010.

[42] WIEN, M. *High Efficiency Video Coding: Coding tools and Specification*. 1 ed. Berlim, Germany, Springer, 2015.

[43] WALLACE, G. K. "The JPEG Still Picture Compression Standard", *IEEE Transactions on Consumer Electronics*, v. 38, n. 1, pp. xviii–xxxiv, February 1992.

[44] LUCAS, L. F. R. *Predictive Coding Algorithms for Lossy Image and Video Compression*. Phd thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, January 2016.

[45] KAMISH, F. *Transform for Prediction Residuals in Video Coding*. Phd thesis, MIT, Massachusetts, USA, 1974.

[46] XU, X., COHEN, R., VETRO, A., et al. "Predictive Coding of Intra Prediction Modes for High Efficiency Video Coding". In: *Picture Coding Symposium*, Krakow, Poland, May 2012.

[47] JAIN, J. R., JAIN, A. K. "Displacement Measurement and Its applications in Interframe Image Coding", *IEEE Transactions on Communications*, v. 29, n. 12, pp. 1799–1808, December 1981.

[48] GIROD, B. "The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video Sequences", *IEEE Journal on Selected Areas in Communications*, v. 5, n. 7, pp. 1140–1154, August 1987.

[49] CHEN, C.-F., PANG, K. K. "Hybrid Coders with Motion Compensation", *Multidimensional Systems and Signal Processing*, v. 3, n. 2, pp. 241–266, May 1992.

[50] HEITHAUSEN, C., BLASER, M., WIEN, M. "Distance Scaling of Higher Order Motion Parameters in an Extension of HEVC". In: *Picture Coding Symposium (PCS)*, Nuremberg, Germany, December 2016.

[51] "HEVC Test Model (HM)". `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/`. Accessed in November, 2017.

[52] WIEGAND, T., SULLIVAN, G. J., BJØNTEGAARD, G., et al. "Overview of the H.264/AVC Video Coding Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 560–576, July 2003.

[53] ITU. *Recommendation ITU-T H.265. High Efficiency Video Coding*. ITU-T Telecommunication Standardization Sector of ITU, April 2015.

[54] JAIN. *Fundamentals of Image Processing*. 1 ed. New Jersey, USA, Prentice Hall, 1990.

[55] AHMED, N., NATARAJAN, T., RAO, K. "Discrete Cosine Transform", *IEEE Transactions on Computers*, v. C-23, n. 1, pp. 90–93, July 1974.

[56] AHMED, N. *How I Came Up with the Discrete Cosine Transform*. NASA 1, 1991.

[57] CHEN, C.-F., PANG, K. K. "The Optimal Transform of Motion-Compensated Frame Difference Images in a Hybrid Coder", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, v. 40, n. 6, pp. 393–397, August 1993.

[58] MALVAR, H. S., HALLAPUNO, A., KARCZEWICZ, M., et al. "Low-complexity Transform and Quantization in H.264/AVC", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 598–603, July 2003.

[59] HUFFMAN, D. A. "A Method for the Construction of Minimum-Redundacy Codes", *Proceedings of the IEEE*, v. 40, n. 9, pp. 1098–1101, September 1952.

[60] MARPE, D., BLATTERMANN, G., HEISING, G., et al. "Video Compression using Context-based Adaptive Arithmetic Coding". In: *International Conference on Image Processing (ICIP)*, Thessaloniki, Greece, August 2001.

[61] MARPE, D., SCHWARZ, H., WIEGAND, T. "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, pp. 620–636, July 2003.

[62] WITTEN, I. H., NEAL, R. M., CLEARY, J. G. "Arithmetic Coding for Data Compression", *Communications of the ACM*, v. 30, n. 6, pp. 520–540, June 1987.

[63] BELL, T. C., CLEARY, J. G., WITTEN, I. H. *Text Compression*. 1 ed. New Jersey, USA, Prentice Hall, 1990.

[64] SZE, V., BUDAGAVI, M. "High Throughput CABAC Entropy Coding in HEVC", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1778–1791, October 2012.

[65] KIM, I. K., MIN, J., LEE, T., et al. "Block Partitioning Structure in the HEVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1697–1706, October 2012.

[66] HEALEY, G. E., KONDEPUDY, R. "Intra Coding of the HEVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1792–1801, December 2012.

[67] LIN, J.-L., CHEN, Y.-W., HUANG, Y.-W., et al. "Motion Vector Coding in the HEVC Standard", *IEEE Journal of Selected Topics in Signal Processing*, v. 7, n. 6, pp. 957–968, December 2013.

[68] SAXENA, A., FERNANDES, F. C. *CE7: Mode-Dependent DCT/DST without 4x4 Full Matrix Multiplication for Intra Prediction, JCT-VC-E125*. Input Document to JCT-VC, March 2011.

[69] BUDAGAVI, M., FULDSETH, A., BJONTEGAARD, G., et al. "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard", *IEEE Journal of Selected Topics in Signal Processing*, v. 7, n. 6, pp. 1029–1041, December 2013.

[70] NORKIN, A., BJONTEGAARD, G., FULDSETH, A., et al. "HEVC Deblocking", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1746–1754, December 2012.

[71] FU, C.-M., ALSHINA, E., ALSHIN, A., et al. "Sample Adaptive Offset in the HEVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 22, n. 12, pp. 1755–1764, December 2012.

[72] WANG, Z., BOVIK, A. C. "Mean Squared Error: Love it or Leave it? A New Look at Signal Fidelity Measures", *IEEE Signal Processing Magazine*, v. 26, n. 1, pp. 98–117, January 2009.

[73] HUYNH-THU, G., GHANBARI, M. "Scope of Validity of PSNR in Image/Video Quality Assessment", *IEEE Electronics Letters*, v. 44, n. 13, pp. 800–8001, June 2008.

[74] WANG, Z., BOVIK, A. C., SHEIKH, H. R., et al. "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, v. 13, n. 4, pp. 600–612, April 2004.

[75] BJONTEGAARD, G. "Calculation of Average PSNR differences between RD-curves". ITU-Telecommunications Standardization Sector, Mar. 2001.

[76] "H.264/AVC Reference Software". `http://iphome.hhi.de/suehring/tml/download/`. Accessed in November, 2017.

[77] AHN, Y. J., HAN, W. J., SIM, D. G. "Study of Decoder Complexity for HEVC and AVC Standards Based on Tool-by-tool Comparison". In: *Proceedings of the SPIE*, San Diego, USA, October 2012.

[78] BOSCH, A., ZISSERMAN, A., , et al. "Scene Classification Using a Hybrid Generative/Discriminative Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 4, pp. 712–727, April 2008.

[79] BROWN, M., LOWE, D. G. "Automatic Panoramic Image Stitching using Invariant Features", *International Journal of Computer Vision*, v. 74, n. 1, pp. 59–73, August 2007.

[80] TUYTELAARS, T., MIKOLAJCZYK, K. "Local Invariant Feature Detectors: A Survey", *Foundations and Trends in Computer Graphics and Vision*, v. 3, n. 3, pp. 177–280, June 2008.

[81] MIKOLAJCZYK, K., SCHMID, C. "Scale and Affine Invariant Interest Point Detectors", *International Journal of Computer Vision*, v. 60, n. 1, pp. 63–86, October 2004.

[82] HARRIS, C., STEPHENS, M. "A Combined Corner and Edge Detector". In: *Fourth Alvey Vision Conference*, January 1988.

[83] MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., et al. "A Comparison of Affine Region Detectors", *International Journal of Computer Vision*, v. 65, n. 1, pp. 43–72, November 2005.

[84] MIKOLAJCZYK, K., SCHMID, C. "Indexing Based on Scale Invariant Interest Points". In: *IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[85] LEUTENEGGER, S., CHLI, M., SIEGWART, R. Y. "BRISK: Binary Robust Invariant Scalable Keypoints". In: *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011.

[86] HEINLY, J., DUNN, E., FRAHM, J.-M. "Comparative Evaluation of Binary Features". In: *Proceedings of the 12th European conference on Computer Vision*, Florence, Italy, October 2013.

[87] WITKIN, A. P. "Scale-Space Filtering". In: *Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983.

[88] LINDEBERG, T. "Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales", *Journal of Applied Statistics*, v. 21, n. 2, pp. 224–270, 1994.

[89] FORSTNER, W., DICKSCHEID, T., SCHINDLER, F. "Detecting Interpretable and Accurate Scale-Invariant Keypoints". In: *International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[90] LINDEBERG, T. "Detecting Salient Blob-like Image Structures and their Scales with a Scale-Space Primal Sketch: A Method for Focus-of-Attention", *International Journal of Computer Vision*, v. 11, n. 3, pp. 283–318, April 1993.

[91] LINDEBERG, T. *Scale-Space Theory in Computer Vision*. 1 ed. Stockholm, Sweden, Springer Science + Business Media Dordrecht, 1994.

[92] LINDEBERG, T. "Feature Detection with Automatic Scale Selection", *International Journal of Computer Vision*, v. 30, n. 2, pp. 79–116, November 1998.

[93] "Open Source Collection of Vision Algorithms (VLFeat)". `http://www.vlfeat.org`. Accessed in January, 2016.

[94] ROSTEN, E., PORTER, R., DRUMMOND, T. "Faster and Better: A Machine Learning Approach to Corner Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 1, pp. 105–119, November 2008.

[95] GAUGLITZ, S., HOLLERER, T., TURK, M. "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking", *International Journal of Computer Vision*, v. 94, n. 3, pp. 335–360, September 2011.

[96] MIKSIK, O., MIKOLAJCZYK, K. "Evaluation of Local Detectors and Descriptors for Fast Feature Matching". In: *International Conference on Pattern Recognition (ICPR)*, Tsukuba, Japan, November 2012.

[97] SCHMID, C., MOHR, R. "Local Greyvalue Invariants for Image Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 5, pp. 530–535, March 1997.

[98] MIKOLAJCZYK, K., SCHMID, C. "A Performance Evaluation of Local Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 10, pp. 1615–1630, October 2005.

[99] KE, Y., SUKTHANKAR, R. "PCA-SIFT: a more Distinctive Representation for Local Image Descriptors". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recoginition*, Washington DC, USA, June 2004.

[100] CALONDER, M., LEPETIT, V., STRECHA, C., et al. "BRIEF: Binary Robust Independent Elementary Features". In: *European Conference on Computer Vision (ECCV)*, Heraklion, Greece, September 2010.

[101] ALAHI, A., ORTIZ, R., VANDERGHEYNST, P. "FREAK: Fast Retina Keypoint". In: *IEEE Conference on Computer Vision and Pattern Recoginition (CVPR)*, Providence, USA, June 2012.

[102] RABIN, J., DELON, J., GOUSSEAU, Y. "A Statistical Approach to the Matching of Local Features". In: *SIAM J. Img. Sci.*, Philadelphia, USA, September 2009.

[103] YANG, W., XU, L., CHEN, X., et al. "Chi-Squared Distance Metric Learning for Histogram Data", *Mathematical Problems in Engineering*, pp. 1–12, March 2015.

[104] KAPLAN, A., AVRAHAM, T., LINDENBAUM, M. "Interpreting the Ratio Criterion for Matching SIFT Descriptors". In: *European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, October 2016.

[105] XIN, X. *Compact Descriptors for Visual Search*. Phd thesis, Northwestern University, Evanston, USA, 2014.

[106] VALLE, E., PICARD, D., CORD, M. "Geometric Consistency Checking for Local-Descriptor based Document Retrieval". In: *ACM Symposium on Document Engineering*, Munich, Germany, September 2009.

[107] SILVA, R. C., PEREIRA, F., SILVA, E. A. B. "Studying the Compression Performance of Video Descriptors". In: *Simpósio Brasileiro de Telecomunicações*, Juiz de Fora, Brazil, September 2015.

[108] ASCENSO, J., BRITES, C., PEREIRA, F. "Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding". In: *Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, July 2005.

[109] PEREZ, P., GANGNET, M., BLAKE, A. "Poisson Image Editing", *ACM Transactions on Graphics (TOG)*, v. 22, n. 3, pp. 313–318, July 2003.

[110] SHEWCHUK, J. R. *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*. Technical Report, August 1994.

[111] GOLUB, G. H., LOAN, C. F. V. *Matrix Computations*. 3 ed. Baltimore, USA, The Johns Hopkins University Press, 1996.

[112] MEYER, C. D. *Matrix Analysis and Applied Linear Algebra*. 1 ed. Philadelphia, USA, Society for Industrial and Applied Mathematics, 2000.

[113] STRANG, G. *Computational Science and Engineering*. 1 ed. Wellesley, USA, Wellesley-Cambridge Press, 2007.

[114] EVERETT, H. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", *Operations Research*, v. 11, n. 3, pp. 399–417, May 1963.

[115] BARBER, C. B. "The Quickhull Algorithm for Convex Hulls", *ACM Transactions on Mathematical Software (TOMS)*, v. 22, n. 4, pp. 469–483, December 1996.

[116] SCHWARZ, H., MARPE, D., WIEGAND, T. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 17, n. 9, pp. 1103–1120, September 2007.

[117] PEREIRA, F., DA SILVA, E. A. B. "Efficient plenoptic imaging representation: Why do we need it?" In: *2016 IEEE International Conference on Multimedia and Expo (ICME)*, Seattle, USA, August 2016.

[118] EBRAHIMI, T., FOESSEL, S., PEREIRA, F., et al. "JPEG Pleno: Toward an Efficient Representation of Visual Reality", *IEEE MultiMedia*, v. 23, n. 4, pp. 14–20, November 2016.

[119] "JPEG Pleno". `https://jpeg.org/items/20150320_pleno_summary.html`. Accessed in February, 2018.

[120] MOREL, J. M., PETRO, A. B., SBERT, C. "Fourier Implementation of Poisson Image Editing", *Pattern Recognition Letters*, v. 33, n. 3, pp. 342–348, February 2012.

[121] MARTINO, J. M. D., FACCIOLO, G., MEINHARDT-LLOPIS, E. "Poisson Image Editing", *Image Processing On Line*, v. 6, n. 1, pp. 300–325, November 2016.