Towards novel inter-prediction methods for image and video compression

Jean Bégaint

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité :Traitement du Signal

Par

Jean BEGAINT

Towards novel inter-prediction methods for image and video compression

Thèse présentée et soutenue à Rennes, le 29 novembre 2018
Unité de recherche : INRIA Rennes - Bretagne Atlantique et Technicolor R&I

Rapporteurs avant soutenance :

David Bull — Professor, University of Bristol, United Kingdom
Markus Flierl — Professor, KTH University, Sweden

Composition du Jury :

Président :	Olivier Deforges	Professor, IETR/INSA, France
Examinateurs :	Franck Galpin	R&D architect, Technicolor, France
	Hervé Jégou	Research Scientist, Facebook, France
	Mathias Wien	Professor, RWTH Aachen, Germany
Dir. de thèse :	Christine Guillemot	Research Director, INRIA, France
Co-dir. de thèse :	Philippe Guillotel	Distinguished Scientist, Technicolor, France

Invité(s)
Dominique Thoreau — Senior Scientist, Technicolor, France

To my family and friends.

Acknowledgements

First of all, I would like to thank my advisors, Franck Galpin, Dominique Thoreau, Philippe Guillotel and Christine Guillemot, for their precious help and guidance during these three years. They were essential to the accomplishment of this thesis.

I am very grateful to the reviewers of this manuscript, David Bull and Markus Flierl, for their comments and suggestions. I would also like to thank the jury members, Olivier Deforges, Hervé Jégou and Mathias Wien for accepting to be part of the jury.

I would like to thank my colleagues from Technicolor and INRIA: permanents, post-docs, PhD students and interns. Many of them have become friends since I started working at Technicolor almost 4 years ago now. In particular, I would like to express my gratitude to the video coding team, for their valuable help and patience.

Thank you to my fellow interns and PhD students: Dmitry, Fatma, Hadrien, Juan, Martin A., Martin E., Matthieu, Mikael, Oriel, Thierry and Salma. Thank you for all these happy moments, babyfoot and tea breaks, lively discussions, diners and trips. Thanks to my friends in Paris and Grenoble, for the nice weekends I spent there.

Finally, I would like to thank my family, especially my parents and brothers, for their continuous support through the years.

Contents

Résumé en Français

Contexte

L'émergence des applications et des services web a donné lieu à un usage croissant des ressources en ligne. En raison de la grande disponibilité des dispositifs de capture vidéo et des nouvelles pratiques liées aux réseaux sociaux, ainsi qu'à l'émergence des services en ligne, les images et les vidéos représentent aujourd'hui une partie importante de données transmises sur internet.

Des milliards d'images sont déjà sauvegardées en ligne, et des centaines de millions y sont téléchargées tous les jours [1]. Ces images sont rarement supprimées et sont souvent dupliquées au niveau des systèmes de fichiers et des centres de traitement de données, afin de limiter des risques de perte et d'améliorer les temps d'accès. Bien que des encodeurs plus récents comme JPEG2000 [2], WebP [3] et BPG [4] ont été proposé, ces images sont généralement toujours encodées avec le standard JPEG [5]. Étant donné le volume considérable de données, les solutions de sauvegarde en ligne pourraient bénéficier des performances de compression des derniers encodeurs. Par ailleurs, des contenus similaires sont très certainement déjà sauvegardées en ligne, cette redondance d'informations pourraient être exploitées pour réduire considérablement les besoins en capacité de stockage. Les systèmes de compression vidéo actuels ont justement été conçus pour exploiter les similarités entre images successives d'une vidéo. En partant d'une base de donnée d'images suffisamment grande, une nouvelle image à enregistrer pourrait alors être encodée à partir d'une ou plusieurs références, déjà sauvegardées en ligne, en utilisant les techniques de prédiction inter des encodeurs vidéos.

Les applications de streaming représentent aujourd'hui plus de 70% de la bande passant mondiale. Avec l'émergence des plateformes de partage en ligne et les services de streaming comme Youtube, Netflix, Hulu, des millions d'heures de vidéos sont transférées chaque jour. Le trafic total des vidéos sur internet devrait atteindre 80% d'ici 2021. Et d'ici là, la bande passante utilisée aura presque doublée. Le volume du trafic de streaming vidéo devrait ainsi passer de 57 exaoctets par mois [1] pour atteindre 159 exaoctets par mois d'ici les quatre prochaines années [6].

En raison de la grande disponibilité des dispositifs de capture vidéo et des nouvelles pratiques liées aux réseaux sociaux, de plus en plus de contenus vidéos sont créés, sauvegardés et téléchargés sur les plateformes en ligne. Pour donner un aspect plus réaliste aux contenus vidéos, de nombreuses améliorations ont été proposées ces dernières années pour étendre les formats vidéos traditionnels. Les films peuvent maintenant être diffusés avec des résolutions ultra hautes définitions (4K), la vitesse du flux vidéo peut dépasser les soixante images par seconde. De nouveaux formats de représentation des couleurs ont aussi été récemment adoptés, tel que l'imagerie à grande dynamique (HDR). Ces nouveaux formats nécessitent des capacités de stockage largement supérieurs aux formats classiques. De plus, des domaines comme la vidéo 360°, la réalité augmentée, la réalité virtuelle, le streaming de jeux vidéos en ligne et le jeu déporté en ligne

[1] Un exaoctet (1 EO) est équivalent à 10^{18} octet, soit 1 million de téraoctets.

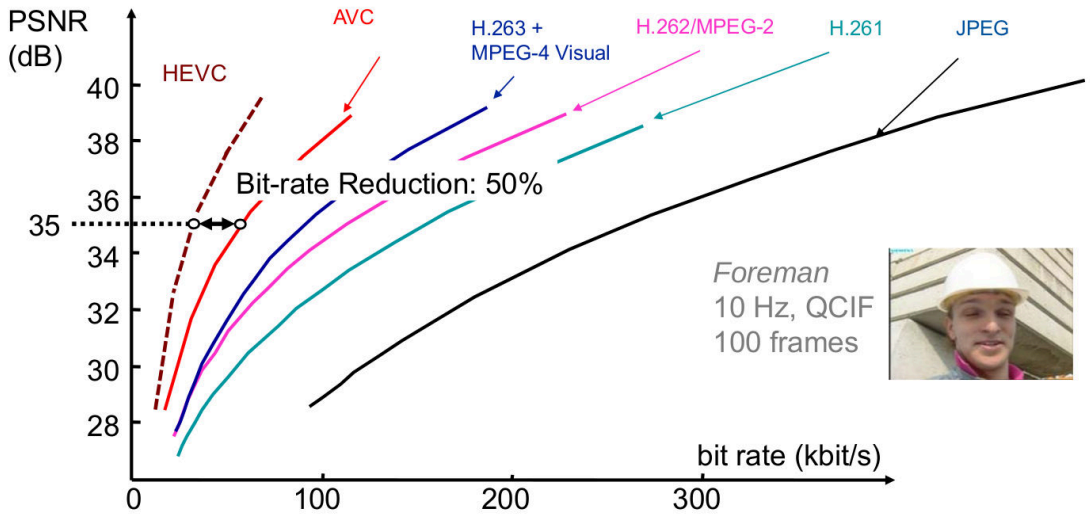


Figure 1: Historique des performances des standards MPEG. *Source:* "Versatile Video Coding towards the next generation of video compression", G. Sullivan and J. Ohm, PCS 2018, San Francisco.

reposent sur des techniques de compression vidéo et devraient avoir une croissance importante dans les prochaines années.

Le dernier standard MPEG de compression video (HEVC) a apporté une amélioration de 50% en performance de codage par rapport à son prédécesseur le standard AVC, et la même amélioration est prévue pour le future standard VVC [7]. Les prévisions du trafic internet de vidéos, les nouveaux usages et les nouvelles formes de contenus soulignent le besoin d'amélioration constante des encodeurs vidéos.

Contributions

Bien que des améliorations pour les encodeurs de compression vidéo actuels soient possibles à plusieurs niveaux, le travail présenté dans cette thèse est principalement axé sur l'étape de prédiction inter images pour la compression d'ensembles d'images et de vidéos.

La première contribution consiste à proposer une méthode de prédiction inter-image pour la compression d'ensembles d'images. En tirant parti des redondances globales et locales entre images similaires, les ensembles d'images peuvent être compressés plus efficacement. La méthode proposée associe une compensation géométrique et photométrique globale avec une compensation locale plus précise.

Une méthode de prédiction inter semi-locale, c'est à dire par région, est ensuite proposée pour compenser les distorsions géométriques et photométriques entre images corrélées. Pour une paire d'image similaires, la méthode est capable d'estimer simultanément plusieurs modèles d'homographies et de compensations photométriques afin de compenser les distorsions entre les images. Des gains significatifs sont démontrés par rapport aux outils de codage d'image traditionnels, ainsi que les méthodes de l'état de l'art sur la compression d'ensemble d'images.

Les encoders vidéos sont traditionnellement conçus pour estimer des déplacement translationnels par blocs de pixels. Ils ne permettre donc pas d'encoder efficacement des mouvements

complexes tels que les zooms de caméra, les mouvements de caméra, les panoramiques. En adaptant la méthode de prédiction inter par région dans un schéma de codec vidéo traditionnel (HEVC), l'efficacité de l'approche est démontrée dans un contexte de compression vidéo classique.

Enfin, la dernière contribution explore les nouvelles approches basées sur les réseaux de neurones profonds. Étant donné les résultats impressionnants obtenus par les réseaux de neurones profonds pour des applications d'interpolation d'images, ces architectures sont étudiées dans un contexte de compression vidéo, notamment en tant que méthode de prédiction inter alternative.

Résumé par chapitre

Le manuscrit est organisé en deux parties. La première partie contient un chapitre d'introduction du domaine de la compression d'image et de vidéo. Ce chapitre est suivi d'une présentation des méthodes de l'état de l'art utilisant des techniques de prédiction inter images. La deuxième et dernière partie de ce document décrit les contributions proposant des nouvelles méthodes de prédiction inter pour la compression d'ensemble d'images et de vidéo.

Chapitre 1: Ce premier chapitre propose une introduction au domaine de la compression d'image et de vidéo. Les bases de la compression d'images et de vidéos y sont présentées. Notamment comment tirer parti des redondances spatiales, visuelles, temporelles et statistiques pour compresser efficacement une vidéo. L'architecture classique d'un encoder vidéo hybride: prédiction, transformation, quantification et codage entropique est présentée. Les éléments clés du dernier standard de compression vidéo (HEVC) sont ensuite introduits, ce standard sera ensuite utilisé comme référence dans tout le manuscrit. Les métriques classiquement utilisés en compression sont également présentées.

Chapter 2: Le deuxième chapitre présente l'état de l'art sur des nouvelles techniques de prédiction inter images. Développées à l'origine pour la compression vidéo, ces techniques ont ensuite été adaptées avec succès pour la compression d'ensembles d'images, tels que les albums photo et les bases de données en ligne. La compensation de mouvement repose traditionnellement sur des mouvements translationnels par blocs, ce qui limite la qualité de la prédiction pour des mouvements complexes comme des zooms ou des rotations. Des modèles de prédiction permettant de compenser ces mouvements complexes ont ensuite été développés et sont présentés ici. Récemment, les réseaux de neurones profonds permettant de faire de l'interpolation d'images consécutives ont obtenu des résultats impressionnants, notamment pour des applications de ralenti de vidéo. Ces méthodes peuvent constituer une voie prometteuse pour améliorer les performances des méthodes actuelles de prédiction inter images. Les principales contributions sur ce sujet sont ainsi présentées dans la dernière section de ce chapitre.

Chapter 3: Le premier chapitre de contribution décrit un modèle de prédiction inter-images combinant une compensation globale et locale. Ce modèle a été développé pour la compression d'ensembles d'images, et exploite les redondances entre images à des niveaux globaux (image) et locaux (blocs de pixels). Pour compenser les distorsions globales entre image similaires, par exemple dues à des différences de points de vue, de caméra, d'objectifs ou d'illumination, une compensations géométrique et photométrique globale est d'abord estimée. Une compensation par bloc de pixels est ensuite effectuée pour compenser les distorsions locales. Cette compensation locale repose sur la théorie *Locally Linear Embedding* (LLE) [8] qui stipule que dans un espace de grande dimension (un manifold), un point peut être représenté comme une combinaison linéaire de ses voisins. Les blocs de pixels d'une image courante peuvent donc être représentés comme une

combinaison linéaire de blocs similaires dans une image de référence. En associant cette méthode de compensation en deux étapes à un schéma de compression vidéo classique, des améliorations significatives sont obtenus par rapport aux méthodes actuelles reposant uniquement sur des encodeurs vidéos.

Chapitre 4: Le deuxième chapitre de contribution présente une méthode de prédiction inter images par région. En segmentant les images en objets ou plans, des mouvements complexes et des distorsions photométriques importantes peuvent être compensés efficacement avec des modèles géométriques et photométriques distincts, tout en assurant une complexité de calcul limitée. L'image à encoder est d'abord segmentée en *super-pixels*, c'est à dire en block de pixels similaires et voisins. Des modèles géométriques paramétriques sont ensuite estimées pour chaque super-pixel obtenu afin de compenser les distortions géométriques par rapport à l'image de référence. L'ensemble des modèles géométriques obtenu est ensuite récursivement sélectionné et ré-estimé afin de ne garder que les modèles les plus significatifs. En combinant les super-pixels segmentés en fonction du raffinement des modèles géométriques, une segmentation par région est obtenu. Finalement, des modèles de compensation photométriques sont estimés pour chaque région obtenue. Des améliorations significatives des performances de compression sont obtenues par rapport aux méthodes de l'état de l'art. La méthode proposée est ensuite adaptée en tant que mode de prédiction inter dans un codeur vidéo. Des réductions importantes de débit sont démontrées, en particulier pour les séquences affichant des mouvements complexes tels que des zooms et des rotations, qui sont difficiles à prédire avec la compensation translationnelle classique par bloc.

Chapitre 5: Un grand nombre de travaux récents de la communauté de vision par ordinateur se sont concentrés sur la résolution des problèmes d'interpolation d'image en utilisant des réseaux de neurones profonds. Ces réseaux permettent d'interpoler une image intermédiaire à partir de deux images successives d'une vidéo. Des résultats prometteurs ont été obtenu par rapport aux techniques classiques de vision par ordinateur. Bien que ces réseaux ciblent principalement des applications de ralenti de vidéo, leurs performances sont étudiées ici dans un cadre de compression vidéo. Plusieurs réseaux d'interpolation d'images ont ainsi été développés et intégré dans un schéma de compression vidéo classique. Dans le cadre de la prédiction temporelle, une nouvelle référence, interpolée par le réseau, est proposée à l'encodeur quand deux images de référence sont disponibles (une référence future et une référence passée). Ces réseaux sont ensuite comparés dans le cadre de la compression vidéo. Des améliorations de débit sont également mesurées par rapport aux méthodes classiques, soulignant le fort potentiel de ces réseaux profonds pour la compression vidéo.

Conclusion

La production et la consommation croissante d'images et de vidéos exigent des innovations constantes des techniques de compression. Grâce aux nouveaux canaux de communication (fibre optique, 4G, 5G), la bande passante disponible auprès des utilisateurs continuent d'augmenter. Néanmoins, cette augmentation nest pas suffisante pour couvrir les besoins liés aux nouveaux usages et aux formats émergents. Par exemple, le modèle historique de la télévision hertzienne est délaissé pour des services en ligne de diffusion de vidéo comme *Netflix*, qui sont de plus en plus utilisés. Un nombre croissant de contenus créés par les utilisateurs sont également téléchargés et visionnés sur les plateformes des réseaux sociaux. Des nouveaux formats tels que la vidéo

360°, la réalité augmentée, la réalité virtuelle, nécessitent une quantité de données toujours plus importante.

Les contributions de cette thèse se sont principalement concentrées sur les techniques de prédiction inter-images pour la compression d'ensembles d'images et la compression de vidéos. Des améliorations significatives de réduction de débit ont été apportées par rapport aux méthodes actuelles. De nouvelles approches basées sur les réseaux de neurones profonds ont également été abordées. De nombreuses améliorations sont possibles afin d'étendre les travaux présentées dans cette thèse. Une étude approfondie sur l'usage et la robustesse des descripteurs locaux pour la compression permettrait notamment d'améliorer les gains. Les modèles paramétriques de compensation de mouvement pourraient également bénéficier de plus de recherche sur la sélection du nombre de degrés de liberté. Concernant les réseaux de neurones profonds, de nombreux travaux restent à effectuer afin de permettre des applications pratiques réelles. Ces développements futurs seront particulièrement intéressants à suivre pour les communautés de la compression vidéo et de la vision par ordinateur.

Chapter 1

Introduction

Context

The emergence of cloud applications and web services has led to an increasing use of online resources. Associated with the large availability of high-end digital cameras in smartphones, as well as the rise of online storage solutions and new social media practices, images and videos represent today a significant part of the internet traffic and cloud storage usage.

Billions of images are already stored in the cloud, and hundreds of millions are uploaded every day [1]. These pictures are rarely deleted and often duplicated across filesystems and datacenters to mitigate data loss risks and improve retrieval times. Although more recent codecs such as JPEG2000 [2], WebP [3] and BPG [4] have been proposed, images are still usually encoded with the classical JPEG [5] codec. Given the considerable amount of data, online storage solutions could benefit from the improved compression performances of latest codecs. Moreover, similar content may already be stored online and this redundancy could be exploited to significantly reduce storage requirements. Video codecs were expressly designed to exploit the similarity between consecutive frames. Given a large enough dataset of images, a new image could then be encoded from a reference, or multiple references, already present in the cloud, by leveraging those video codecs inter-prediction tools.

Video streaming applications account today for more than 70% of the world internet traffic bandwidth. With the emergence of media sharing websites and streaming services such as Netflix, YouTube and Hulu, millions of hours of videos are now streamed every day, and the traffic is expected to reach 80% of the total internet traffic by 2021. By then the total IP traffic will have almost doubled, video streaming is predicted to grow from 57 exabytes/month[1] to 159 exabytes/month in the next 4 years [6].

Due to the large availability of video cameras and new social media practices, more and more video contents are being created, stored and uploaded in the cloud. To improve the realism of the content, several enhancements have been made to traditional video formats over the last years. Ultra High Definition (4K) and high frame-rate (HFR) videos are now available. New color representation such as wide color gamut (WCG) and high dynamic range (HDR) were also recently adopted. These new formats require large additional amount of data compared to traditional videos. Besides, emerging fields such as 360° videos, light fields, augmented reality (AR), virtual reality (VR), video games streaming and cloud-gaming services usage heavily rely on video compression technologies and are expected to grow significantly in the following years.

[1] One exabyte (1 EB) equals to 10^{18} bytes, or 1 million terabytes.

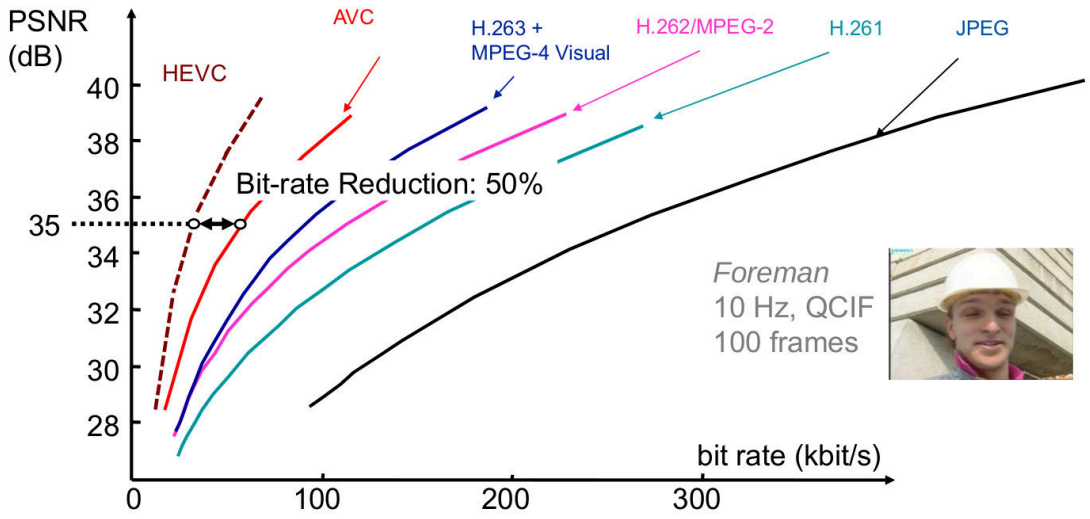


Figure 1.1: Performance history of MPEG standard generations. *Source:* "Versatile Video Coding towards the next generation of video compression", G. Sullivan and J. Ohm, PCS 2018, San Francisco.

Although the latest MPEG video compression (HEVC) standard improved by 50% the compression efficiency over the previous AVC standard (see Figure 1.1), and the same improvement is targeted for the future VVC [7] standard, the forecasted traffic, emerging usages and new content forms indicate that further improvements in video compression are needed.

To leverage correlations between similar frames, video codecs use inter-prediction algorithms which estimate and compensate the differences between frames. Such differences are usually due to camera or objects motions, and illumination changes. Inter-prediction methods traditionally rely on block-based motion vectors, estimated via block-matching algorithms. However, motion vectors are restricted to model block-based translational displacements, which limits the performances for complex motions such as zooms and camera shakes. In this thesis, we study novel inter-prediction approaches, with larger capacities of generalization in order to handle complex distortions. Both image sets compression and video compression applications are targeted.

Contributions

(i) The first contribution consists in proposing additional inter-prediction methods for image sets compression. By leveraging global and local redundancies between similar images, image sets can be compressed more efficiently. Although significant gains can be obtained, the high complexity of the local prediction limits the usability of the proposed method.

(ii) A semi-local, or region-based, inter-prediction method is then proposed to compensate geometric and photometric distortions between correlated frames. For a given pair of similar frame, the method is able to estimate simultaneously multiple homography and photometric models to accurately compensate distortions between frames. Significant gains are demonstrated over traditional image coding tools and state-of-the-art methods for compressing image sets.

Considering that video codecs are traditionally designed to handle rigid, block-based, two-dimensional displacements, they do not perform well for encoding efficiently specific real world motion types such as camera zooms, shakes, pans. By inserting our region-based inter-prediction approach into a traditional video codec, the efficiency of the proposed approached is demonstrated compared to the state-of-the-art HEVC video codec.

(iii) Finally, the last contribution explores new approaches based on deep neural networks. Given the impressive results obtained by deep convolutional network for frame interpolation applications, theses architectures are studied for use in a video compression codec as an additional inter-prediction technique.

Structure of the thesis

The manuscript is organized in two parts. The first part contains a chapter introducing the image and video compression domain, followed by a presentation of the state-of-the-art methods leveraging inter-prediction techniques. The second and last part of this document describes our contributions on novel inter-prediction methods for image set and video compression.

Chapter 1: This chapter provides an introduction to the image and video compression domain. The basics of compression are presented, as well as key elements of the latest High Efficiency Video Coding (HEVC) standard, which will be used as a reference baseline throughout the manuscript. The metrics classically used in compression are also presented.

Chapter 2: The second chapter presents the state-of-the-art on novel inter-prediction techniques. Originally developed for the video compression domain, such techniques have been successfully used in image sets compression tasks such as photo-album and cloud-based image compression. Differing from the classical translational motion compensation, more complex motion models have also been presented for video compression. Recent work leveraging deep neural networks for frame interpolation have obtained striking results for slow-motion applications and may constitute a promising direction for improving current inter-prediction performances. The main contributions on this subject are presented in the last section of the chapter.

Chapter 3: The first contribution chapter describes a combined global and local inter-prediction model for image-sets compression. By exploiting global and local inter-image redundancies, this model is able to compensate complex distortions between similar images. Significant rate-distortion performance improvements are measured against classical video compression tools.

Chapter 4: The second contribution chapter presents a region-based method for inter-prediction. By segmenting input frames into regions, objects or planes, large motions and photometric distortions can be compensated efficiently with distinct geometric and photometric models, while ensuring a limited complexity. Significant performance improvements are obtained compared to state-of-the-art methods. The proposed method is then adapted as an inter-prediction mode in a video codec. Bit-rate reductions are demonstrated, especially for sequences displaying complex real world motion such as zooms and rotations.

Chapter 5: The third contributions chapter focused on deep learning based methods. A large number of recent works from the computer vision community have focused on solving frame interpolation problems using deep neural networks. By inputting consecutive frames into a deep

convolutional network, intermediate frame(s) can be interpolated directly. Promising results have been obtained in comparison to classical computer vision techniques. Although these networks were designed to solve slow motion applications, their relevance for video compression is studied here. Indeed a network able to interpolate intermediate frames could be particularly useful for frame inter-prediction from past and future reference frames.

Part I

Context and state-of-the-art

Chapter 2

Background in image and video compression

2.1 Principles of image and video compression

Image and video contents represent the most significant part of the digital data. To reduce storage requirements and improve transmission times, redundancies within image and video signals can be exploited to compress more efficiently the content. However, these redundancies can not be directly expressed by statistical distribution models and require specific techniques designed for these signal characteristics.

2.1.1 Redundancies removal

A video consists in a succession of frames. Each individual frame can be viewed as an individual static image, and as such video and image compression techniques share some overlap. Four different types of redundancies are present in a video, the three redundancies found in an image and the additional temporal one:

- **Spatial redundancy**: neighbouring pixels, or even blocks, usually have similar values as they may be part of the same object or background with a consistent texture. This redundancy can be exploited by predicting pixel values from neighbouring pixels (intra-prediction). When transforming the signal with the Discrete Cosine Transform (DCT), the coefficients can also be ordered according to their contribution to the signal.
- **Visual redundancy**: the Human Vision System (HVS) is imperfect and particularly insensitive to partial loss of data. Some pixel values can thus be changed to compress the image more efficiently without compromising too much the perceptual quality. This is especially exploited when quantizing coefficients in the transform domain, where high frequencies can be removed. Another common technique is to represent the image in the YCbCr color space, where the luminance and the red/blue chrominances are separate channels. The chrominance channels are usually down-sampled by 4 (*i.e.* by 2 in each vertical and horizontal directions), as the HVS is more sensitive to changes to the luminance channel.
- **Temporal redundancy**: due to the continuous nature of a video, adjacent frames often display the same scene. Visual differences are mainly caused by camera motions or objects

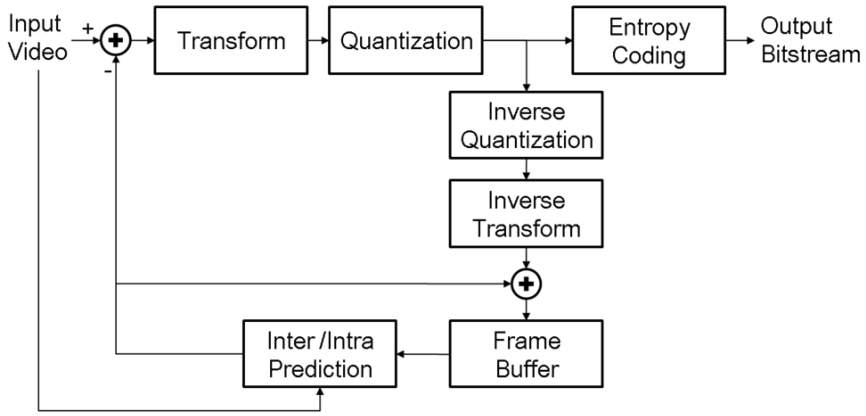


Figure 2.1: Illustration of the hybrid video coding framework [10].

moving in the scene. This high redundancy can be leveraged by estimating and compensating accurately the different motions. The camera and the object displacements can be modeled by parametric motion models, such as local block translations.

- **Statistics redundancy**: in order to store the pixels of the image, the coding information (modes, coefficients, etc ...) are described as a succession of symbols (or set of bits). The distribution of these symbols is not random, and as such displays some correlation that can be exploited by source coding like arithmetic coding algorithms. Remaining statistical redundancies can be further exploited by using entropy coding such as CABAC [9].

Video coding schemes exploit these four types of redundancies to provide better compression performances. It is important to note that most video codecs feature lossy compression techniques, *e.g.* the decoded signal does not match exactly the original source signal. Some distortion is introduced by the encoding process, especially during the quantization step of the transform coefficients. One common artifact of lossy compression is the blocking artifact usually seen on images and videos encoded at low bit-rates. Without lossy compression, the bit-stream sizes of media contents would not be suitable for most storage and transmission use cases. When encoding at medium to high bit-rates, the artifacts of lossy compression become difficult to detect for the casual observer due to the imperfections of the human vision system. Compression schemes usually operate in two modes: they either minimize the distortion for a given bit-rate or minimize the bit-rate for a given distortion. Constant bit-rate and variable bit-rate encoding are generally respectively used for streaming and storage applications.

The same classical framework has been in used for the last decades to design video codecs, and is still pertinent today. It features both a prediction/motion compensation step and a transform step. It is thus known as the hybrid video coding framework (Figure 2.1). The principal building blocks of the hybrid video coding framework are represented in Figure 2.1. The inter/intra prediction step is designed to leverage the spatial and temporal redundancies. The transform and quantization steps make use of the spatial and visual redundancies. Finally the statistics redundancy is reduced by the entropy coding block.

The main improvements in recent codecs may be explained by new partitioning topologies. The block partitioning algorithms have changed from a fix square block based partitioning to novel quad-trees and binary trees partitioning, allowing complex non-linear predictions and transformations to compress more efficiently. At the detriment of the complexity on the encoder side, due to the large number of combinatorial possibilities to be tested.

2.1.2 Predictive coding

Due to spatial and temporal redundancies, neighbouring pixels and frames may share similar content. Predictive coding techniques have thus been developed to use these redundancies by estimating current block values from neighbouring blocks or frames. A residual is then usually computed, transformed, quantized and entropy coded. When decoding the bit-stream, the residual will be added to the prediction to compensate for the inaccuracies.

The frames of the sequence to encode are first divided into blocks. Each block is then encoded using spatial (intra) or temporal (inter) prediction. Spacial predictions are based on the neighbouring blocks, while inter prediction leverages the previously encoded/decoded frames which are likely to be correlated to the current frame. Encoders traditionally try both intra and inter predictions with their different modes and select the best prediction mode based on a combine rate-distortion criterion:

$$J = D + \lambda \cdot R \tag{2.1}$$

where D is the distortion between the original and reconstructed block, usually measured by the Mean Square Error (MSE/l_2 norm). R is the bit-rate, *i.e.* the number of bits required to encode the block. The λ parameter is set by the encoder to set the trade-off between the distortion and the bit-rate. For example in HEVC, λ can be set as function of the quantization parameter. This loop over the possible prediction modes is named the Rate Distortion Optimization (RDO) process. As this step can only be performed on the encoder side, the original frame being unknown on the decoder side, specific syntax is added in the bit-stream to signal the chosen mode and its parameters. The residual, *i.e.* the difference between the reconstructed and the original block, is then transformed, quantized and encoded in the bit-stream. The encoder also decodes the previously encoded block as it will be needed for the predictions of the next blocks to encode. The same decoding operation must be performed on both the encoder and decoder side, as the decoder will blindly predict the block from the syntax coded in the bit-stream, so the block used as reference for the prediction must have the same values on both sides.

2.2 Overview of HEVC

This section presents an overview of the state-of-the-art HEVC/H.265 video codec [11, 12]. The HEVC standard, or H.265/MPEG-H Part 2, was finalized by the Joint Collaborative Team on Video Coding (JCT-VC) in 2013. HEVC was designed to bring a bit-rate reduction of 50% compared to its predecessor, the Advanced Video Coding AVC/H.264 codec [13], finalized in 2003. The aim of this section is not to present exhaustively all the features of HEVC but rather to detail key characteristics of HEVC that will later be referred to in this document.

2.2.1 Quad-tree structure

Like its predecessors, HEVC processed the frames in a block-wise manner. To adapt the encoding to the content, the frames are divided recursively into multiple blocks of pixels. The HEVC standard introduced a quad-tree structure for the block partitioning. Each upper block in the tree structure thus has four block children of the same size.

The HEVC standard defines four different types of blocks in the quad-tree structure:

- Coding Tree Unit (CTU): is the largest block structure in HEVC. When building the quad-tree, the frame is first divided into CTUs of 64x64 pixels.

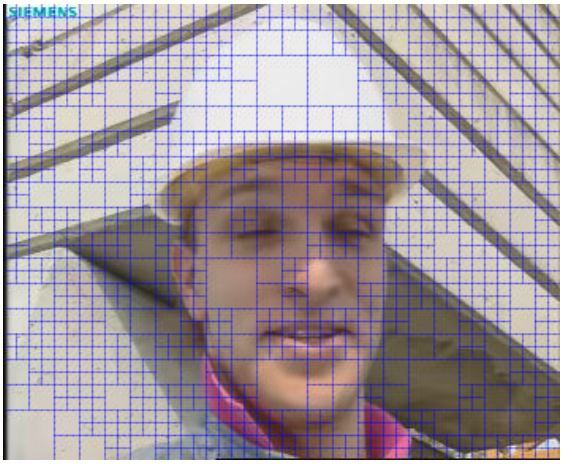


Figure 2.2: Example of the HEVC quad-tree partitioning on the "Foreman" sequence. Only the Coding Unit (CU) partitioning is represented here.

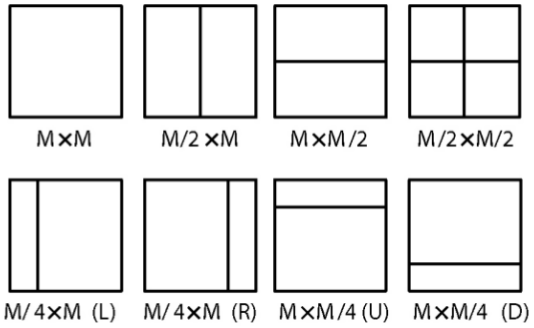


Figure 2.3: The eight possible PU partition schemes.

- Coding Unit (CU): the previously obtained CTUs can be divided into 4 CUs. Each CUs can also be divided recursively into 4 smaller CUs. Up to three levels of recursion are allowed in the HEVC standard, from 64x64 pixels down to 8x8 pixels. The choice of the prediction mode, intra or inter prediction, is performed at the CU level. CUs are processed in a Z-scanning order, or zig-zag order: from the top left CUs to the bottom right ones, going right to left.

- Prediction Unit (PU): each CU can be divided into multiple PUs. The prediction information, motion vector for inter or mode index for intra, are estimated and stored at the PU level. A CU can contain up to four PUs. Several partitioning schemes are available and differ from the previous quad-tree partitioning. For the intra mode, only squared PUs are available, so an intra CU may only have one or four PUs. For the inter mode, eight configurations are defined as rectangular PUs are allowed: two squared PUs, three vertical rectangular PUs, and three horizontal rectangular PUs.

- Transform Unit (TU): each CU is also divided into one or several TU. The transform and quantization step are performed on the TU level. Each TU can be split into multiple

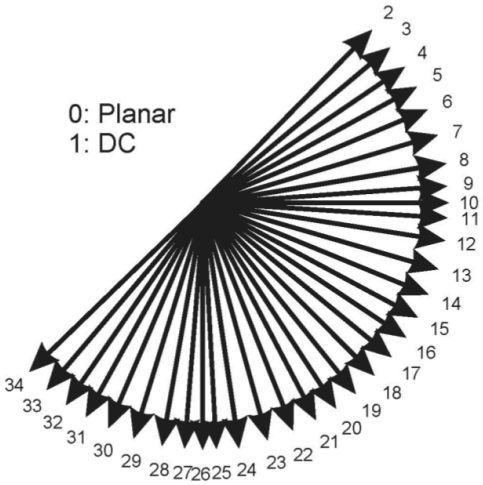


Figure 2.4: DC, planar and the 33 directional intra prediction modes in HEVC.

smaller TUs in a quad-tree structure. TU sizes ranges from 32x32 to 4x4, and are also processed in a zigzag order.

2.2.2 Intra prediction

The intra prediction mode is designed to exploit the spacial redundancy within the current frame. The intra prediction relies on the neighbouring reconstructed blocks values. These reconstructed blocks have been previously encoded and decoded, so their pixel values will also be available during the decoding process. As the blocks are processed in a zigzag scan, pixels on top and left of the current block can be used. As multiple block sizes are possible with HEVC, pixels on the bottom left and top right may also be retrieved for prediction in some cases. When processing the first blocks of the frames, no neighbors are available for the prediction. A padding operation is thus performed beforehand.

Several methods, or modes, can be used to predict the current block values from the neighbouring pixels. The HEVC standard defines 35 intra prediction modes: DC, planar and 33 directional modes (see Figure 2.4). The DC prediction is defined as the average of the neighbouring pixels. The planar mode consists in a multi-directional prediction, horizontal and vertical, from the neighbouring pixels. The directional modes are used to predict the current pixel values by extending the neighbouring pixels in a given direction. The index of the mode is chosen and transmitted at the PU level. The directional modes are represented in Figure 2.4. The neighbouring pixels border used for a directional mode is depicted in Figure 2.5.

The chroma components (Cb and Cr) can only be predicted from five modes: planar, DC, horizontal, vertical and Direct Mode (DM). Prediction with the Direct Mode is performed by using the same mode selected from the luma component. This mode relies on the strong correlation between the luma and chroma components.

2.2.3 Inter prediction

The inter prediction is designed to leverage the temporal redundancy between consecutive frames. The basic idea is to use previously encoded and decoded frames as references to encode the current

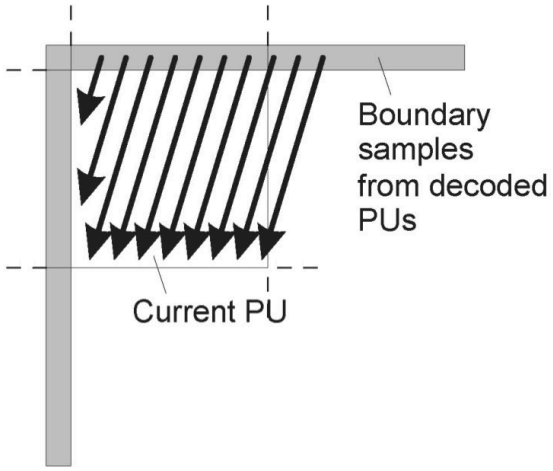


Figure 2.5: Intra prediction pixel samples available from the neighbouring reconstructed blocks.

frame. Multiple reference frames can be used to encode the current frame. The HEVC standard defines three types of frames:

- I frame: an I frame is only encoded using the intra prediction mode. Since the intra prediction does not use the temporal redundancy, the I frames have a high coding cost and represent a significant part of the total bit-rate of an encoded sequence. Yet I frames are necessary to be able to seek in time when watching a video sequence. Otherwise the decoder would have to decode all the previous reference frames just to decode the current frame. I frames are also inserted when there is no correlation between consecutive frame, for example during a scene cut in a movie.
- P frame: a P frame can be coded using intra prediction or inter prediction from past reference frames.
- B frame: a B frame can be coded using both past and future reference frames, and intra prediction. Compared to the I and P frames, B frames have a significantly lower bit-rate. To be able to use future frames for prediction the encoding order differs from the temporal order. B predictive frames were introduced in H.264/AVC [14].

The HEVC standard divides a sequence into multiple Group Of Pictures (GOP). A GOP may contain I, P and B frames. The GOP structure defines the encoding order of the frames. GOPs are encoded successively as there may exists coding dependencies between consecutive GOPs. A classical GOP structure in HEVC is the hierarchical GOP structure. The first frame is encoded as an I frame, the last frame as a P frame (predicted from the first frame or other past frames from past GOPs), the intermediate frames are encoded recursively as B frames. An example is shown in Figure 2.6.

The inter prediction is performed by finding translational motion vectors for each prediction unit. The motion vectors are defined with a quarter pixel accuracy to obtain better prediction. The reference fame index and the motion vector parameters (dx, dy) are encoded in the bit-stream, the decoder will then be able to perform the same prediction. In order to reduce the size required to encode the motion vector parameters, a prediction is also performed. A motion vector prediction is obtained from the previously neighbouring PUs encoded with inter-prediction, or

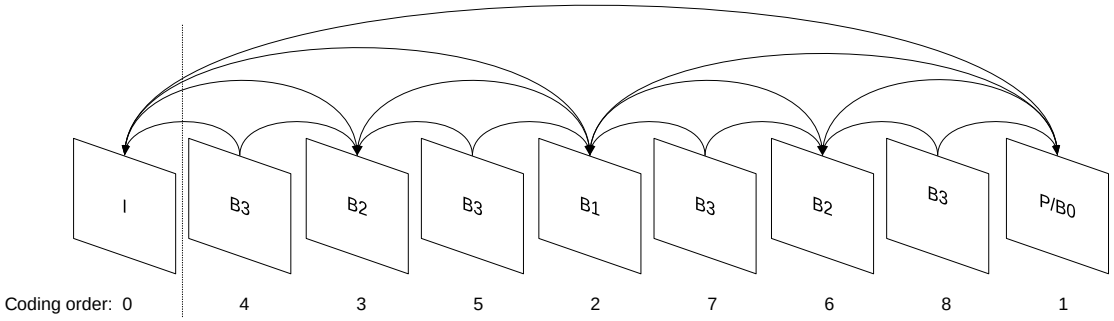


Figure 2.6: A traditional hierarchical GOP structure. P and B frames can be predicted from multiple reconstructed reference frames.

from motion vectors from other frames, only the difference (residual) between the predicted motion vector and the estimated one is actually stored in the bit-stream.

The HEVC standard defines two variants of inter prediction: "merge" and "skip". For these two modes, only the motion vector prediction is performed, there is no motion compensation step. Up to five motion vector candidates are collected from neighbouring PUs, only the index of the selected one is stored in the bit-stream. Compared to the merge mode, the skip mode does not encode the block residual values. The reconstructed block is the same as the predicted one. Both these modes require less side information to transmit to the decoder and are computationally less expensive than classical inter prediction as they do not require the motion estimation. However, they rely on high temporal correlations as they are less accurate that the inter mode.

2.2.4 Transform and quantization

Like most video codecs, HEVC defines the quantization operation in the transform domain. Each TU is transformed by computing a two-dimensional discrete transform. The HEVC standard uses the Discrete Cosine Transform (DCT) [15] to represent the block in the frequency domain. Transform basis are defined for TU blocks of size 4x4 up to 32x32. For the luma TU of size 4x4, the Discrete Sine Transform (DST) is used instead and it was shown to perform better in this case. For a signal x of size N, the DCT (DCT-II version) is defined as:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad \text{with } k \in [0, N-1] \tag{2.2}$$

So for a two-dimensional signal I of size $N \times N$, the DCT-II is defined as:

$$\mathcal{I}(u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j) \cos\left[\frac{\pi}{N}\left(i+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{N}\left(j+\frac{1}{2}\right)v\right] \tag{2.3}$$

The basis functions for the 8x8 DCT transform are represented in Figure 2.7. Scaling can be performed on the coefficients to make the matrix orthogonal, which greatly simplifies the invert DCT transform. This scaling is usually paired with a quantization factor in compression applications.

The DCT is traditionally used in image and video compression as it compacts the signal energy on the top left coefficients. The first coefficient $\mathcal{I}(0,0)$ is known as the DC coefficient and

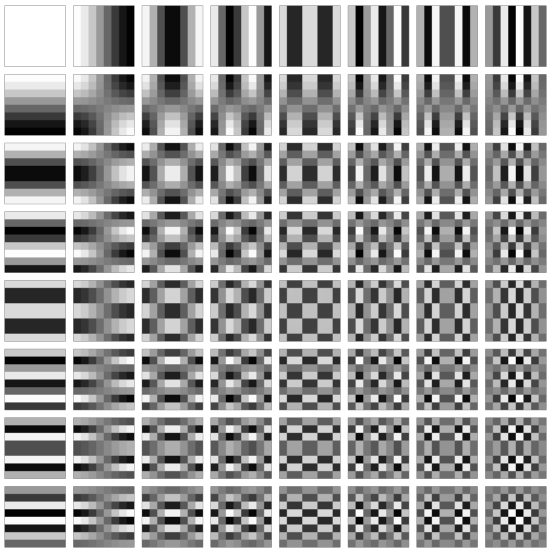


Figure 2.7: The two-dimensional DCT frequencies for a block of size 8x8.

corresponds to the average value of the block pixels. The DCT is also separable, which means it can be computed less expensively in two steps. First by computing a one dimensional DCT horizontally, then vertically on the previously obtained coefficients.

Once the intra and inter prediction have been performed, the residual coefficients are transformed with the DCT and then quantized. The quantization is performed uniformly in HEVC, *i.e.* a fixed quantization step is defined for all the block coefficients. The step value is derived from the Quantization Parameter (QP) that takes values in the range $[0, 51]$. For high bit-depth (greater than 8 bits), the QP can take negative values for encoding with a high quality.

The DCT coefficients are represented with 16 bit integer values in HEVC. The coefficients are shifted after each one-dimensional transform to avoid overflows. The final coefficients are scaled based on the QP.

2.2.5 CABAC entropy coding

The entropy coding of the bit-stream information is performed by the Context Adaptive Binary Arithmetic Coder (CABAC). All the syntax, motion vectors, intra mode indices and quantized transformed residual coefficients are coded by the CABAC algorithm. CABAC is a binary coder, as such all the information must be represented as a sequence of bits. It relies on arithmetic coding to efficiently encode the information. Arithmetic coding algorithms choose the most efficient binary representation of a value by using its probabilities.

The probabilities of the symbols are updated continuously when encoding with CABAC. Each information to be encoded with CABAC has a probability initialized via a given context, this probability is then updated each time the same context is used. CABAC context are defined for most of the information to be encoded. The context enables the CABAC encoder to exploit correlations between the previously encoded flags. For example the prediction mode, or the intra prediction index are coded with CABAC and their probabilities are derived for the previously encoded CUs.

Non binary data, such as motion vectors or transform coefficients, need to be binarized.

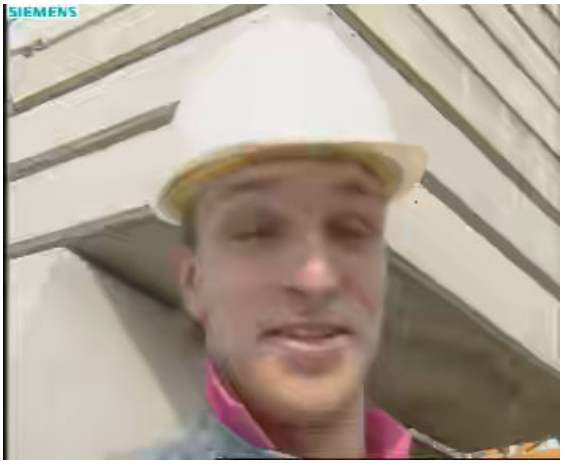


Figure 2.8: Artifacts on "Foreman" sequence, encoded with $QP = 42$.

Integer values can be easily converted to binary bins via a fixed length representation.

2.2.6 In-loop filtering

Due to the lossy nature of the HEVC compression artifacts are present in the reconstructed frames. When encoding at high QP (low bit-rate), these artifacts can easily be seen. The block wise encoding process is responsible for the "blocky" artifacts, the most current image/video compression artifacts. Ringing artifacts may also be present due to the use of the DCT, distortions can be seen along strong edges.

The HEVC standard defines two filters to remove these artifacts: the deblocking filter and the Sample Adaptive Offset (SAO) to prevent ringing artifacts. These filters are applied in the coding loop, hence the term "in-loop filtering". SAO also includes a mode to correct banding artifacts (usually visible in low-texture regions such as skies). These filters can thus also improve the prediction, as the filtered frames will later be used as predictions. The filters parameters need to be transmitted to the decoder. One downside of in-loop filtering is that a filtering process improving the perceived quality does not necessarily improves the references for the prediction.

2.3 Metrics for compression

Metrics are required to compare coding performances between different codes or prediction methods. Several metrics can be used to measure the quality of a compressed video at a given bit-rate, either describing the quantitative distortion or the visual quality.

2.3.1 PSNR

The Peak Signal to Noise Ratio (PSNR) measures the distortion of a retrieved signal compared to its original version. The PSNR can be used to assess the fidelity between the original frames and the reconstructed ones. The PSNR is computed pixel-wise:

$$PSNR = 10 \cdot \log_{10}\left(\frac{d^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{d}{\sqrt{MSE}}\right) \tag{2.4}$$

(a) Uncompressed (b) $Q = 90$, PSNR 36.23dB, SSIM 0.962 (c) $Q = 30$, PSNR 30.91dB, SSIM 0.896 (d) $Q = 10$, PSNR 27.53, SSIM 0.820

Figure 2.9: Example PSNR and SSIM values for the test image *Lena* encoded with JPEG (*libjpeg* [16]) at various quality levels (Q).

with d the maximum possible value for a pixel (*e.g.* 255 for a 8 bits image), and the Mean Square Error (MSE) is defined for two single channel images I and J of size $m \times n$ as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - J(i,j))^2 \tag{2.5}$$

When comparing color frames, in the RGB or YCbCr color spaces for example, the MSE or PSNR is either averaged on each channel, or is reported separately for each channel. Although the PSNR is not a direct measure of the visual quality of an image, and differs from the way the human vision system evaluates an image quality, its has been widely adopted as the default quality metric for the development of image and video codecs. In the case of lossy image and video compression, values usually range from 20dB to 50dB (higher is better).

2.3.2 SSIM

The Structural Similarity (SSIM) was designed to evaluate the perceptual quality differences between two images. It was first proposed in 2004 by Wang *et al.* [17] and has since been adopted by the image processing community. The SSIM is not computed pixel-wise, but rather on sub-windows (x, y) of the images, and is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.6}$$

with μ_x and μ_y the respective average of the x and y windows, (σ_x, σ_y) the corresponding variances, σ_{xy} the covariance of the two windows, and $(c1, c2)$ two fixed variables to stabilize the division by avoiding numerical issues.

Example values of PSNR and SSIM metrics are provided in Figure 2.9.

2.3.3 Bjøntegaard metric

When comparing two codec versions, differences can be measured on the distortion or bit-rate levels. Either by observing the distortion improvements for a given bit-rate, or measuring the bit-rate reductions for a fixed distortion.

Bjøntegaard *et al.* introduced in [18] a simple framework to simplify the comparison between two prediction methods at multiple bit-rate levels. They proposed two metrics: the BD-rate and

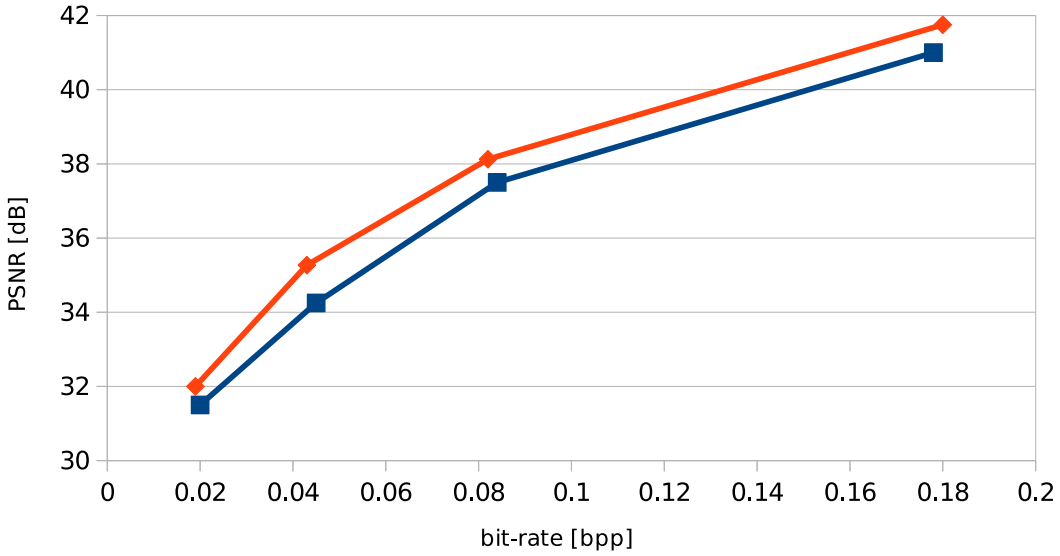


Figure 2.10: Example Rate-Distortion curve. Bit-rates and PSNRs are measured for four different quantization levels, *e.g.* 22, 27, 32 and 37 for HEVC.

the BD-PSNR. Which respectively describe the average bit-rate or PSNR differences between two encoding methods. These metrics are computed for four points from a Rate Distortion curve (RD-curve). RD-curves enable an easy visual comparison of two compression methods. The curve on the "top left" represents the most efficient compression method: the decompressed quality is the highest for a fixed bit-rate, and similarly the bit-rate is the lowest for a fixed distortion. An example RD-curve is shown in Figure 2.10.

The BD-PSNR measures the average PSNR difference between two RD-curves. The BD-PSNR is calculated using third degree polynomials over logarithmic bit-rates and PSNRs data points:

$$BD - PSNR = \frac{1}{(r_H - r_L)} \int_{r_l}^{r_h} (D_2(r) - D_1(r))dr \tag{2.7}$$

$$BD - rate = \frac{1}{(D_H - D_L)} \int_{D_l}^{D_h} (r_2 - r_1)dD \tag{2.8}$$

with $r_h = log(R_h)$, $r_L = log(R_L)$ being the high and low boundary values of the output bit-range. D_1 and D_2 are the two RD-curves considered for comparison. The BD-rate is expressed in percentage, the BD-PSNR in decibel (dB). As it describes a bit-rate difference, the BD-rate has negative values when there is an improvement, *i.e.* a bit-rate reduction.

Chapter 3

Predicting images from images

Inter-prediction algorithms have first been designed for video compression. Predictive coding by estimating and compensating the camera and objects motion between consecutive frames has been successfully proven in video codecs. However videos are not the only type of media displaying high redundancy between consecutive frames. Image sets from medical, satellite data, photo albums or even image cloud databases also require a significant amount storage and may benefit from improved inter prediction tools developed for compressing videos.

In this chapter the literature focusing on image prediction is introduced. Works on image sets compression tasks are first presented, with a focus on photo album and cloud-based image compression. We will then describe novel improvements to the classical motion compensation scheme used in video codecs. Recent approaches leveraging the efficiency of deep neural networks for video compression are then introduced. Finally, frame prediction methods based on deep learning are presented.

3.1 Image sets compression

3.1.1 Overview and motivation

The first works on image sets compression came from the medical and satellite imaging domains. In these applications images are typically highly correlated as the observation conditions slightly vary. Early solutions involve encoding the target image as its difference to a Representative Signal (RS) of the images set. The RS can be a reference image, an average or a median of all the images in the set [19]. However, approaches involving representative signals are not robust to geometric transformations (*e.g.* displacements, cropping, focal length) or photometric transformations (*e.g.* illumination changes). Under such distortions, the RS is not correlated anymore to each image. The prediction errors are then too significant to be efficiently encoded via residual coding.

Dynamic data-sets, with additions, deletions or modifications of images, are also an issue for these methods since the RS has to be recomputed for each modification of the sets, in order to preserve the efficiency of the prediction and the compression. To exploit the similarities between photos with less correlations, such as pictures from photo albums or cloud-based image databases, more robust methods have thus been considered, especially by exploring tools from the video compression domain.

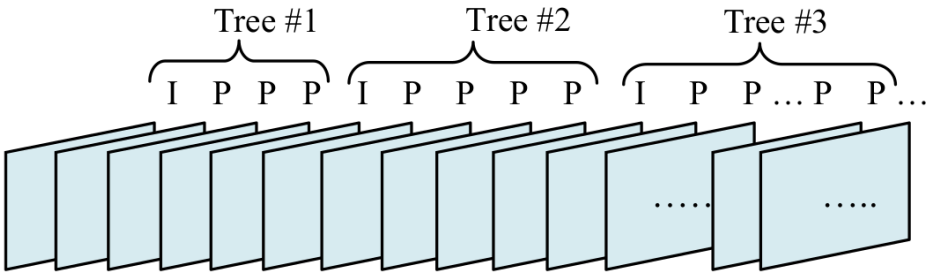


Figure 3.1: Photo album encoded as a pseudo-video sequence. *Source*: Zou *et al.* [20].

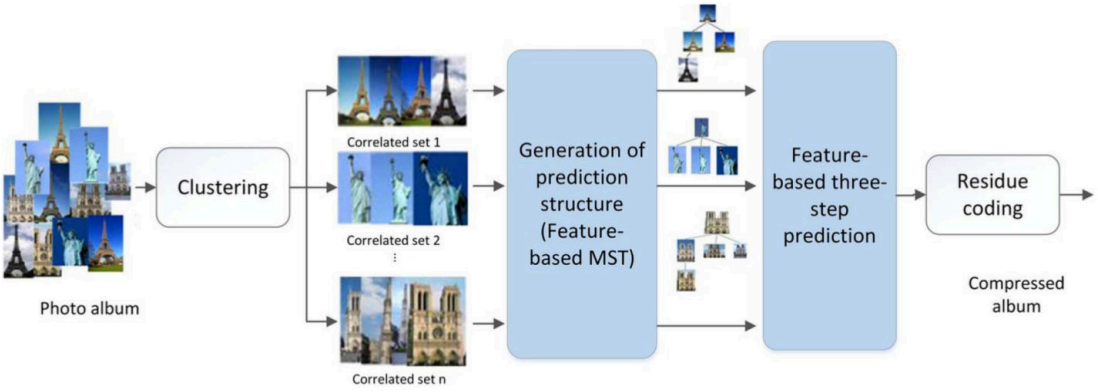


Figure 3.2: Photo album compression scheme of Shi *et al.* [21]. The images are first clustered by scenes based on clustered local features. Global geometric compensations are then derived from the locally matched features via RANSAC [22] and the geometric distortions are corrected. Finally, local disparities are compensated with residual coding by encoding the tree as a pseudo-video sequence. *Source*: Shi *et al.* [21].

3.1.2 Photo-album compression

To exploit the redundancies in a photo album, Zou *et al.* proposed in [20] to organize images from an album into a tree structure, and then to encode it as a video sequence. A graph structure is first determined by order of similarity between the images. Correlations are measured with the Sum of Square Differences (SSD). An image tree is then obtained from the graph via a Minimum Spanning Tree approach (MST) and encoded with HEVC, with a Group Of Pictures (GOP) of one I frame (the tree root) and n following P frames (the leaves). The scanning is performed via a depth-first search algorithm, meaning the lowest leaves are explored before going upwards. A maximum depth of the tree is imposed in order to limit the image retrieval time (the random access). The constructed pseudo-sequence is depicted in Figure 3.1. The authors obtain an overall improvement of 75% over JPEG. However, this method relies on the SSD for measuring the correlation, which is not robust to geometric and illumination changes. In addition, accessing a random image requires prior decoding of several images and increases the loading time. Moreover, video encoders have not been designed to cope with variations in terms of focal length, viewpoint, illumination, usually encountered in sets of images.

As a pixel based measure is not robust to describe the correlation between images, Shi *et al.* proposed in [21] an approach based on local feature descriptors. They introduced a three steps

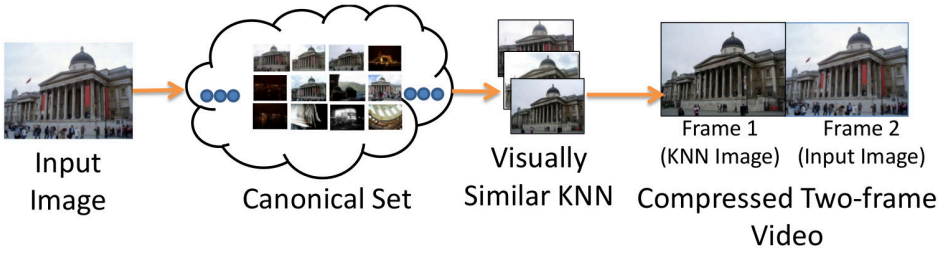


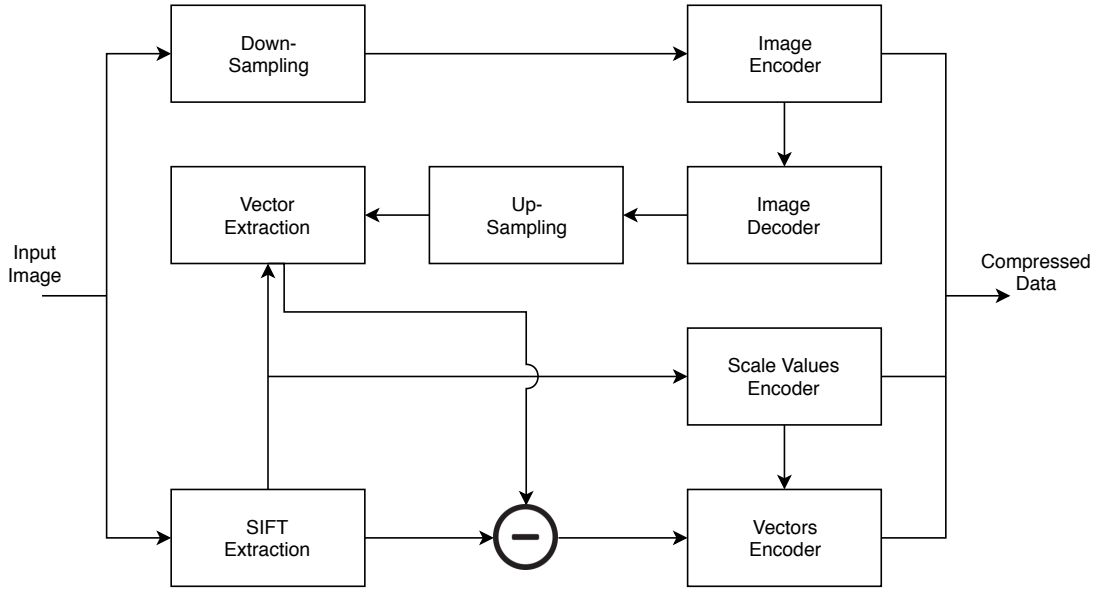
Figure 3.3: Overview of Perra & Frahm compression scheme [24]. The image to be encoded is matched to visually similar images via a k-nearest neighbour search in a large canonical database. From each image in the closest k results two-frame videos are encoded. The reference image yielding the best compression ratio is then selected, and the resulting bit-stream is stored. *Source*: Perra & Frahm [24].

method to reduce inter-image redundancies. A feature-based multi-model approach is first used to compensate geometric transformations between images. Then, a photometric transformation is applied to account for illumination changes between the references and the target image. Finally, a Block Matching Compensation (BMC) is performed to compensate for remaining local disparities. The method is represented in Figure 3.2. To evaluate the geometric transformations, a content-based feature matching is first performed by using SIFT local feature descriptors [23]. The matching between images is based on the correlation between groups of descriptors instead of pixel values. A K-means algorithm is applied to cluster SIFT descriptors and organizes the images into correlated sets. The images are placed in a graph, the weights are computed as the distance between matched SIFT feature vectors. The prediction structure is obtained by converting the graph into a MST. The number of transformations and their parameters are then derived, and the geometric transformation is then estimated via the RANSAC algorithm [22]. A feature-based photometric compensation is proposed to compensate for illumination changes. Finally, block motion compensation is used to account for local disparities, which are not compensated by the global geometric and photometric compensations. This method outperforms JPEG by reducing the bit-stream size by a factor of 10, while maintaining the same quality.

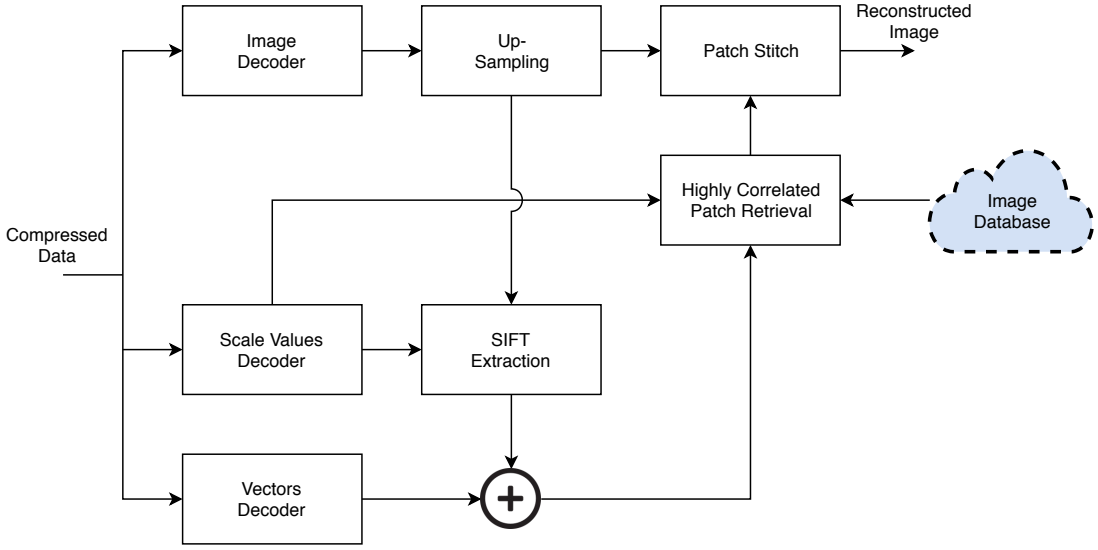
3.1.3 Cloud-based image compression

Considering the millions of images available in the cloud, it is likely that for a given image, another highly similar image can be found in a large database [24]. Perra *et al.* thus proposed to take advantage of available online datasets and the inter-coding performance of HEVC to compress image pairs, and introduced a novel approach with low computational cost. To find correlated images, global feature descriptors are used. A GIST descriptor [25] is computed from the current image and then reduced to a 512 bits representation for faster processing. GIST descriptors have been selected as they are as efficient as SIFT in this context [26], and with a lower computational cost. A nearest neighbour search (K-NN) is then performed to retrieve the most correlated image from the dataset. An HEVC inter-coding is finally applied with the reference image as an I frame and the query image as a P frame. The proposed compression scheme is depicted in Figure 3.3. This method provides fast operations, suitable for online applications, and produces an average reduction of size by a 74% factor with a canonical set of 13 million images, compared to JPEG.

Additional methods have been developed to deal with sets of images with larger disparities. As such, Yue *et al.* proposed in [27] to encode an image from its down-sampled version and



(a) "Encoder"



(b) "Decoder"

Figure 3.4: Overview of Yue *et al.* compression scheme [27].

local feature descriptors. The descriptors are used to retrieve correlated images from the cloud and identify corresponding patches. As an image can have thousands of SIFT descriptors [23], the total size of feature vectors can exceed the image size. The SIFT descriptors of the current image are thus encoded from the SIFT descriptors extracted from the down-sampled version of the image. Only the compressed descriptors and the encoded down-sampled image are then sent to the cloud. Once the data has been decompressed, the image can be reconstructed. First, highly correlated patches are retrieved from the cloud. The transformation between a pair of patches (retrieved and up-sampled) is estimated by applying the RANSAC algorithm on the descriptors. Finally, the patch stitching is guided by the up-sampled decompressed image. The proposed encoding method is represented in Figure 3.4a, and the decoding process in Figure 3.4b. This method achieves an average 1885:1 compression rate, and yields a better subjective quality than JPEG and HEVC intra-coding. However, this method has some limitations. For some images, one may not find sufficiently correlated images in the cloud. Complex images can also be too difficult to reconstruct accurately. In this case, the authors then proposed to extract the complex parts of the image and encode them with classical image compression tools. Furthermore, this method requires high computational power to perform all the operations. Although good visual results and an impressive compression ratio can be obtained, this technique might not reconstruct faithfully the original image due to the use of sparse local feature, and the absence of residual coding.

It is also worth pointing out that similar registration techniques based on local features extracted from correlated images have also been used successfully for image super-resolution [28–33] and image denoising [34, 35] tasks.

Recently, Zhang *et al.* presented [36] a novel prediction method based on dense correspondences. They proposed to compensate geometric and photometric distortions on a 256×256 pixels block basis. For each 256×256 unit, an homography model is estimated from matched pixels via the RANSAC algorithm. A luminance adjustment is then performed on the Y channel by estimating a scale and an offset parameters. The estimated parameters are stored in the bit-stream as side information. By using dense pixel to pixel correspondences [37] in local units instead of local descriptors, the parametric estimation of the geometric models and the luminance compensation is shown to be more robust to local disparities.

3.2 Inter prediction for video compression

3.2.1 Geometric motion models

The efficiency of video compression tools heavily relies on their ability to reduce the temporal redundancy between consecutive frames. Video codecs are primarily designed assuming that rigid, block-based, two-dimensional displacements are suitable models to describe the motion taking place in a scene. However, translational models are not sufficient to deal with real world motion types such as camera zoom, shake, pan, ... Such complex motions are currently handled by splitting large objects into multiple coding blocks compensated with translational motion models. This requires more side information to code the block splitting tree and produces inaccurate predictions, which consequently result in costly residues.

Using more complex transformation models has long been investigated by the video compression community. Early attempts were proposed for MPEG-4 [38] to apply homographic global motion compensation to sprites [39, 40] and for an associated global and local compensation [41] in H.263 [42]. These approaches were discarded at that time in favor of translational models and dense block partitioning, both for coding performance and complexity reasons. Recent works have demonstrated that coding improvements could still be achieved by using global homographic motion models in current state-of-the-art video codecs.

In the ongoing work to improve the compression efficiency of the HEVC [11] codec local, block-based, affine modes were proposed [43, 44] to the Joint Video Exploration Team (JVET) [45]. Translation motion vectors of neighborhood blocks are used to derive affine motion parameters for the current block. Thus no additional information or motion model parameters estimation is required. Chen *et al.* extended the method to support non-square block partitioning and multiple reference frames. Significant gains on the targeted affine sequences and good results on the common test condition (CTC) sequences [46] were demonstrated.

Recently, support for global (frame-based) and local (block-based) homography models has also been proposed and integrated in the emerging AV1 codec from the Alliance for Open Media [47, 48]. The global motion model is estimated from matched FAST [49] local keypoints. The matching is performed based on a fixed euclidean distance threshold and the correlation between the current and candidate points. The global motion model is then determined via the RANSAC algorithm. Several model types are considered: identity, translation, similarity, affine or homography. The best model parameters are quantized and stored at the frame level for each reference frame. The authors also introduced a warped local motion model. From the previously estimated translational motion vectors of the current block, and its causal neighborhood blocks using the same reference frame, they estimate a local warping model that can be uses as a predictor. If this warped predictor is selected, the model is signaled in the bit-stream at the block level (thus only for single-referenced inter-blocks with neighborhood blocks using the same reference frame). Parker *et al.* demonstrated the effectiveness of such motion prediction tools on videos with complex or steep camera motion.

3.2.2 Deep neural networks for video compression

Following recent successes in using deep neural networks for numerous challenging computer vision tasks, Deep Neural Networks (DNN) have also been proposed in the domain of video compression. Such architectures can be useful at different stages of the encoding process: before, during and after the rate-distortion optimization loop.

For example, deep neural networks have been successfully trained to reduce the encoding time by predicting the quad-tree splitting decision [50, 51], a complex and costly decision process due to the large combinatorial possibilities to be tried. New prediction modes have also

been proposed, *e.g.* by predicting the current block from its neighbours [52, 53]. Novel motion compensation methods based on deep learning have also been introduced [54]. Finally the post processing step can also benefit from such approaches [55, 56], especially since CNN have been proven quite performant for super-resolution and restoration applications [57, 58].

However these approaches do not tackle the challenging issue of designing an end-to-end deep convolutional network for video compression, *i.e.* a network designed to single-handedly compress a video. Such architectures have been recently studied for static image coding. The recent works from Ballé *et al.* [59, 60], Rippel and Bourdev [61], and Toderici *et al.* [62] have proven that deep networks can be trained for solving image compression tasks.

Rippel and Bourdev [61] proposed an auto-encoder network leveraging a pyramidal decomposition, and an adaptive coding and regularization module, trained with an adversarial loss to optimize the visual quality at low bit-rates. The auto-encoder extracts features independently at different scales, these features maps are then concatenated and processed to extract joint information across the scales. The final features are quantized and binary coded via an adaptive arithmetic coding module. The training is performed with a reconstruction and an adversarial loss, similar to GAN [63] networks. A discriminator network is thus simultaneously trained to classify between the reconstructed image and the original target. The auto-encoder network and the discriminator networks are trained alternately, depending on the discriminator accuracy.

Toderici *et al.* [62] introduced recurrent neural network (RNN) architectures for full resolution image compression. By using a recurrent convolutional neural network architecture for the entropy coder, dependencies between image patches can be captured and leveraged. A convolutional RNN architecture is also used for the encoder and decoder networks.

Ballé *et al.* published in [60] a variational auto-encoder (VAE) for image compression. Their model rely on an hyperprior to capture spatial redundancies in the VAE latent representation. This hyperprior is similar to side information. The concept is similar to classical image encoders which use side information to signal prediction modes or partitioning in order to optimize the encoding for a given image. In the proposed solution, the network is trained end to end to optimize both the reconstruction quality of the image and the amount of side information.

Current performances already exceed classical image codecs like JPEG [5] and JPEG2000 [2], and can almost reach the performances of state-of-the-art codecs such as HEVC [11] in PSNR.

To the best of our knowledge, the first end-to-end deep video codec was recently proposed by Wu *et al.* [64]. By building on the proposed deep image coding network of Toderici *et al.* [62] and the literature on frame interpolation and extrapolation [65–67], they introduced a fully deep neural architecture for video compression. Similarly to the classical hierarchical video coding approach, they statically encode key-frames and interpolate the remaining frames hierarchically, starting from the key-frames as input references. A notable contribution is the extraction of additional side information to guide the interpolation process, especially for large time steps. Current performances match the H.264/AVC [13] video codec on the provided bit-rates, and highlight the strong potential of such architectures for video compression in the years to come.

Several novel approaches have also been recently proposed to improve the current tooling of the classical hybrid video coding framework. The Joint Video Experts Team (JVET), working on HEVC successor, the Versatile Video Coding (VVC) codec, has seen several submissions based on deep learning methods. These proposals target the in-loop and post-filtering filters, intra coding, and also predicting the block partitioning for faster encoding [1].

[1] http://phenix.int-evry.fr/jvet/

3.3 Deep learning for frame interpolation

Frame interpolation tasks have been recently covered by numerous contributions relying on deep neural networks. Such architectures have been shown to obtain significant results compared to classical solutions. From a triplet of temporally consecutive frames, the objective of these networks is to interpolate the intermediate frame from the past and future input reference frames.

Numerous and diverse approaches have been developed to solve this task. Early methods leveraged fully convolutional networks to predict the targeted frame. Following works focused on working in a transform domain, resulting in less blurry results that in the pixel domain. Recent approaches proposed to first estimate the motion between the frames, and then interpolate the intermediate frame from the input frames and the estimated pixels motion. Alternatively, adaptive convolutional approaches replace these two-steps approaches by directly prediction local convolution kernels. The intermediate image is then synthesized from the inputs frames convolved by the predicted kernels.

Although this section focuses on the frame interpolation problem, these architectures can also be trained for frame extrapolation, *i.e.* to predict the next frame(s) given several past reference frames.

3.3.1 Fully convolutional approaches

The first approaches to focus on frame interpolation problems relied on fully convolutional networks. Two input images are fed into a network made of fully convolutional layers and activations functions, and one prediction image is obtained. When optimizing the prediction, the mean square error (MSE), or l_2 loss, is often used to describe the difference to the ground truth. However results are highly blurry. As multiple solutions are possible for each pixel, the l_2 loss will optimize for the average of all these equally possible modes, resulting in the loss of textures.

In [68], Mathieu *et al.* proposed several solutions to improve the sharpness. They first introduced a hierarchical multi-scale architecture to estimate the interpolated frame at multiple spatial resolutions. The inputs are first down-sampled to the lowest resolution. Then the prediction is successively estimated, up-sampled and refined at each scale level. They also proposed to use a gradient loss function. By simply taking the neighbouring pixel intensities differences as a gradient function, more sharpened output can be obtained. Finally they introduced an adversarial term to the total loss. Following the results from Goodfellow *et al.* [63], they trained a discriminator network to classify between interpolated frames and ground truth frames. The adversarial loss term increases with the discriminator network ability to classify correctly a generated image. Both networks are trained alternately until convergence. Finally, the authors outlined that better results can be obtained when using a l_1 norm instead of a l_2 norm to measure the difference to the ground truth. Indeed, the l_1 loss will drive the network to choose the median of the possible pixel values instead of the average value, the results will thus be less blurry.

3.3.2 Transform-based predictions

Amersfoort *et al.* proposed in [69] an unsupervised transformation based model for next frame prediction in video-sequences. Although they present results for frame extrapolation, their architectures could be adapted for frame interpolation and is thus presented in this section. Instead of working in the pixel domain to predict the next frame for given past frames, they worked in a transform domain. Given a set of past reference frames, they estimated the affine transform between each pair of consecutive frames, and then train a network to output the next

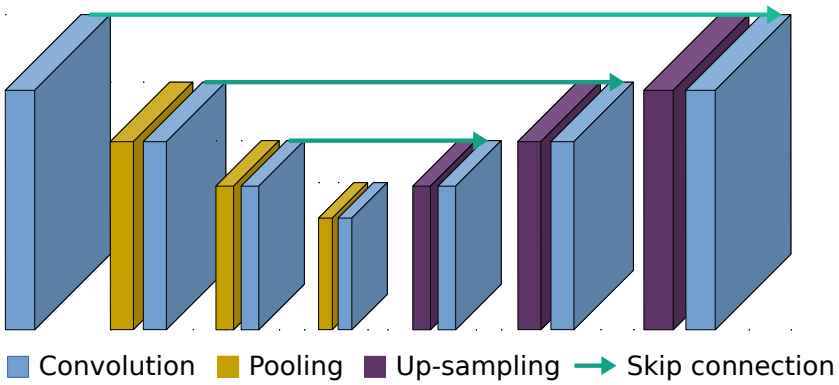


Figure 3.5: Convolutional auto-encoder architecture. The skip connections allow the network to keep spatial details and high level information. Pooling layers are usually average or max pooling layers. The up-sampling layer can either be a simple bilinear up-sampling or a deconvolution operation. Trainable layers are followed by an activation function such as tanh or ReLU [72]. Batch normalization [73] may also be used before the activation function.

affine transformation. The next frame can then be synthesized by interpolating the predicted affine transformation with the last past frame. The network is trained with a MSE loss on the transform coefficients. Affine transforms are estimated on overlapping patches to predict small motions. By operating in the transform domain their approach is able to obtain sharper results than traditional approaches in the pixel domain. Their architecture can predict up to 4 next frames. Unfortunately the approach is evaluated on predicting plausible frames instead of a quantitative evaluation compared to the ground truth.

3.3.3 Interpolation via optical flow

A classical technique to estimate motion at the pixel level is to use an optical flow algorithm. For each pixel, a translational vector $v = (dx, dy)$ is estimated to match the displacement of the given pixel to its position in the second frame. Several deep learning approaches have had impressive results on classic optical flow benchmarks, *e.g.* FlowNet2 [70], PWC-NET [71]. A recurring problem when learning deep networks is the availability or creation of a synthetic dataset of sufficient quality. In this case a database of groundtruth motion from real videos would be required to train a network in a supervised manner.

To circumvent the lack of sufficient databases, Liu *et al.* [67] were the first to introduce an architecture with an unsupervised learning of the optical flow. Similarly to the classical fully convolutional approaches, their *Deep Voxel Flow* network is trained end-to-end to interpolate an intermediate frame from two consecutive input frames. However the last layer of the network is a non-trainable interpolation function interpolating the previous layers output and the input frames. The loss function is still computed directly on the targeted frame reconstruction and the ground truth. By constraining the network in such a way, it will learn by itself to estimate a form of optical flow and a selection mask for the trilinear interpolation layer. The proposed network architecture is a classical convolutional auto-encoder with skip connection, as represented in Figure 3.5. The network outputs a *voxel flow* which represents the per pixel displacement (dx, dy) and a temporal mask dt weighting the trilinear blending. For input images of size

$H \times W$, the voxel flow has a size of $H \times W \times 3$. Assuming the motion is temporally linear between the two input frames I_0 and I_1, their respective flow can by computed as:

$$F_0 = (x - dx, y - dy), \text{ and } F_1 = (x + dx, y + dy) \tag{3.1}$$

The interpolated intermediate frame can then be obtained as:

$$I_{0.5} = W \circ \mathcal{B}(I_0, F_0) + (1 - W) \circ \mathcal{B}(I_1, F_1) \tag{3.2}$$

with $\mathcal{B}$ the bilinear interpolation function and W the temporal mask composed of all dt where $W_{i,j} = [0, 1] \; \forall i, j$.

To cope with larger motion displacements, they also proposed a stacked multi-scale variant. The inputs are processed independently at different resolutions, then the output features of each scale are concatenated and passed through additional convolutional layers to output the final refined flow. Several approaches have followed this work, introducing different architecture and loss functions.

Amersfoort *et al.* proposed in [74] a more traditional hierarchical multi-scale approach. At each scale, the optical flow (dx, dy) and the temporal mask (dt) are jointly refined. A first coarse flow estimation is computed at the lowest scale. Then, the upper scales recursively refine the estimation. A flow residual is computed at each intermediate scale from the up-sampled estimation of the previous scale. To correct local artifacts from the trilinear interpolation, a final convolutional refinement is performed on the pixels value of the interpolated output. To train the network, they used a combined l_1 norm. A $l1$-loss is computed on the refined output, on the raw output, and on the interpolated output at each scale. By using the $l1$-loss at each scale, they are able to train successfully the network. Without the multi-scale supervision, the optical flow estimation is an ill posed problem as there is multiple possible flow solutions per scale. To preserve the texture and the sharpness of the interpolation, perceptual terms are also added to compose the final loss function. A loss based on the difference in features space is applied from a pre-trained *vgg-16* network [75]. An adversarial network is also trained to discriminate interpolated output. Their approach is fast and is shown to perform better than the current state-of-the-art with only 160K trainable weights, compared to the 9M weights of the multi-scale DeepVoxelFlow approach [67].

Latest frame interpolation works leveraged a bi-directional estimation of the optical flow, followed by a synthesis network [76, 77]. In [77], the authors proposed a context-based interpolation method. The optical flow is estimated in both temporal directions between the two reference frames with a multi-scale pyramidal estimation network (PWC-Net [71]). A spatial context is estimated by extracting an intermediate layer output of a pre-trained ResNet-18 [78]. The estimated flows are scaled to the desired temporal interpolation step and used to warp the references and the contexts. The warpings are then sent to a synthesis network based on a GridNet [79] architecture to generate the targeted frame. By processing the inputs simultaneously at different scales, the GridNet architecture is able to synthesize a sharp image, leveraging low level information. It is worth noting that the use of the contextual information improves the interpolation quality by more than 1db (PSNR).

In [76], Jiang *et al.* proposed a novel bi-directional architecture based on two U-Net [80] networks. A U-Net network is a type of convolutional auto-encoder, as the one previously shown in Figure 3.5, with multiple consecutive convolution layers at each scale. A bi-directional flow is estimated by the first U-Net network. For a given interpolation time step $t \; \forall t \in]0, 1[$, intermediate flows are estimated by interpolating directly the flow values. The previously estimated flows and

the warped references are then sent to the second U-Net which refines the flows and computes a temporal mask. The targeted frame is then classically interpolated from the refined flows, the temporal mask and the reference frames.

Recently, Xue *et al.* presented a task-oriented optical flow framework [81]. They trained a deep neural network for optical flow estimation and then fine-tuned it for three different applications: frame interpolation, denoising and super-resolution. Each of these tasks is performed by a separate deep convolutional network. They demonstrated that learning, or at least fine-tuning, an optical flow network for a specific task can outperforms traditional approaches.

3.3.4 Adaptive convolution

Niklaus *et al.* proposed in [66] an adaptive convolution approach to video frame interpolation (*AdaConv*). Instead of relying on a motion estimation then pixel synthesis two-steps approach, the authors proposed to implement the pixel synthesis as a convolution over the two input frames. They trained a deep neural network to predict a spatially adaptive convolution kernel for each pixel. For each pixel, parameters for a 2D convolutional kernel are predicted, then the input frames are convolved at the current pixel location to predict its interpolated value. Their method is able to handle occlusions, blur and brightness change. However the large kernel to be estimated for each pixel is computationally expensive, as they need a 51×51 pixel kernels to handle large motions. This drawback is addressed in their following work [65] in which they introduced separable convolutions (*SepConv*) using 1D kernels for faster processing.

3.3.5 Frame interpolation for video compression

Several deep learning architectures have been proposed to solve frame interpolation and extrapolation tasks. Significant qualitative results have been obtained with these networks that can generate visually pleasing slow-motion effects. In the context of video compression, these methods could be used as an alternate inter-prediction mode by interpolating predictors from past and future reference frames. However it is unclear how these network would perform for video compression applications, especially with larger temporal step between references as found in a classical group of pictures. Moreover these networks are trained to minimize the distortion between the generated frame and the groundtruth (for example with the l_1-norm), whereas video codecs operate with a rate-distortion trade-off, *i.e.* both the bit-rate and the distortion need to be minimized.

Part II

Contributions

Chapter 4

Global and local prediction models

4.1 Introduction

Inter-coding of images is traditionally used for video compression, where the temporal redundancy is reduced by encoding consecutive frames from previous frames used as references. Solutions have been proposed to leverage the inter-prediction tools of video codecs to encode similar images as pseudo-video sequences [20, 24]. However, when considering similar images from a image set, differences between images do not occur only due to camera or objects motions. Differences can be characterized by geometric transformations (*e.g.* translations, rotations, zoom) or photometric transformations (*e.g.* illumination disparities, gamma changes). As such, novel prediction tools are able to compensate such distortions are required.

This chapter describes a novel cloud-based compression scheme based on a data dimensionality-reduction technique. Assuming a correlated image can be found in the cloud, a global geometric and photometric compensation is first performed to account for the geometric transformations and illumination disparities. A locally weighted template-matching based prediction is then performed to predict a reference image. Finally, the target image is inter-encoded with the predicted reference image in HEVC. A crucial aspect of this scheme is the accurate prediction that can not be achieved by coarser global compensation mechanisms based on sparse local features. The proposed method can thus leverage the advanced inter-coding tools of video encoders such as HEVC. The rest of this chapter is organized as follows. Section 4.2 gives an overview of the proposed compression scheme. Section 4.3 and Section 4.4 respectively detail the global and local compensation methods. Experiments are reported in Section 4.5. Section 4.6 concludes this chapter.

4.2 Overview of the proposed scheme

When considering an image I_c to be encoded, a reference image I_r is first retrieved from the cloud with the help of a classic Content Based Image Retrieval (CBIR) system. A predicted image I_p is then constructed from the inter-images correlations. The current image I_c is then coded in reference to the predicted image I_p with HEVC. The main purpose here is to predict I_c with sufficient accuracy to leverage the efficient inter-prediction modes of HEVC and thus reduce the bit-rate required to encode I_c. In this way, a new image uploaded online can be encoded more

efficiently from a similar image retrieved from the cloud than with static picture coding tools. For the purpose of explanation, only pairs of similar images are considered in this chapter, but the proposed scheme could also be adapted for larger sets of similar images.

The prediction mechanism proposed in this chapter relies on a global compensation which is associated to a local prediction mechanism, to compensate accurately the multiple disparities that may occurred between the images. The global compensation is performed to adjust the geometric and photometric distortions on the global level. A template-matching based method is then applied on the pixels to compensate local distortions at the blocks level.

4.3 Global compensation mechanism

The geometric deformation between two similar images can be described as an homography, which models the perspective transformation between two planes. The homography matrix is represented in Equation 4.1, where $h_{11}, h_{22}, \ldots, h_{32}$ are the homography coefficients.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{4.1}$$

To compute the transformation model one widely used approach consists in estimating the transformation from matched local descriptors. Local descriptors are first extracted on both images and then matched in pairs. We used the SURF descriptors [82] to describe the local features in our implementation. A transformation model is then estimated from the matched descriptors by applying the RANSAC [22] algorithm. The process is shown in Figure 4.1. Feature descriptors are resistant (to some extent) to geometric and photometric differences, and as such are more robust than pixel values for the homography estimation.

To compensate for the photometric disparities, we used the global color mapping algorithm proposed in [37]. A correction method [83] for panorama and an histogram reshaping technique [84] were first considered but the adjustments suffer from some inconsistencies if some colors are not present in both images. Also the method did not handle correctly saturation. The global mapping algorithm consists in fitting each RGB channel on a constrained monotonic curve and then apply a linear transform to prevent saturation. The saturation adjustment is modeled with a multiplying coefficient s on the gray line. The matrix C (Equation 4.2) is used to solve the saturation problem with several weighting models for w_r, w_g, w_b (uniform and YUV) and the best result is kept. This method present several advantages as it is robust to saturation, non-linear tone adjustments and can handle colors that do not overlap in the image pairs [37].

$$C = \begin{bmatrix} s - w_r & w_g & w_b \\ w_r & s - w_g & w_b \\ w_r & w_g & s - w_b \end{bmatrix} \tag{4.2}$$

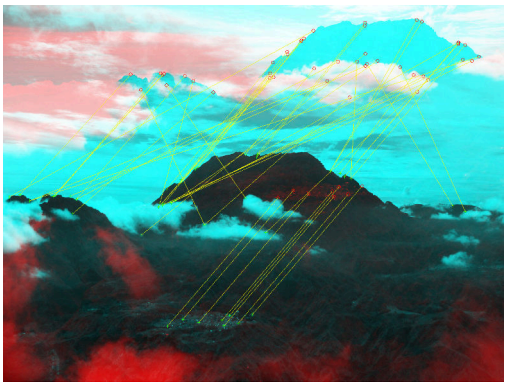
The homography matrix and the photometric adjustment parameters need to be encoded and included in the bit-stream for further decoding. The decoder will then be able to perform the same transformation operations. The parameters are encoded as 16-bit half-floats.

4.4 Locally weighted template-matching based prediction

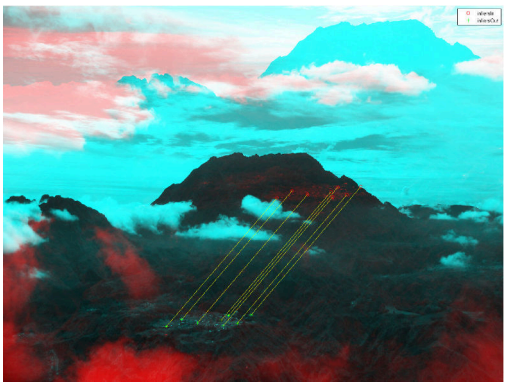
In this section, we describe an inter prediction algorithm based on Locally Linear Embedding (LLE) [8]. Specifically we used the locally weighted template-matching prediction algorithm introduced in [85] as an inter prediction mode for video compression such as in H.265/HEVC.

(a) Current image

(b) Reference image



(c) Keypoints matching



(d) Inliers selection

(e) Compensated image

Figure 4.1: Illustration of the global compensation process.

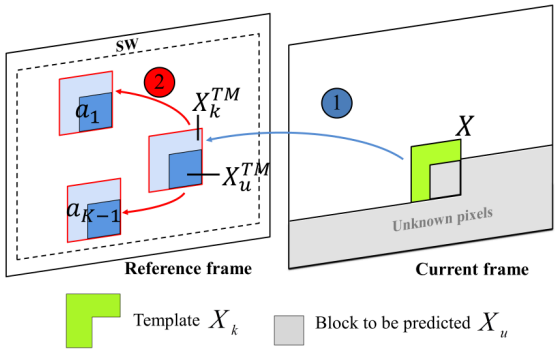


Figure 4.2: K-NN search in the reference image using template-matching.

The Template Matching (TM) method is derived from the block matching technique used in block motion compensation. Instead of looking for the most correlated blocks in the reference frame, TM exploits the correlation between a block and its neighboring pixels, the template (Figure 4.2). As opposed to block matching, there is usually no need to transmit a motion vector since the template of the current block is known on the decoder side (the templates being composed of the previously decoded neighboring blocks).

LLE, a data dimensionality reduction technique, has been proposed for intra-prediction in [86, 87] as an extension to TM. The main idea of the LLE algorithm consists in representing a data point of a high-dimensional space as a linear approximation of its neighbors.

The inter prediction method presented here relies on the assumption that each block in the current frame can be approximated from correlated blocks retrieved in a similar frame. Each block in the current image I_c is thus reconstructed as a linear combination of its K Nearest Neighbors (K-NN) retrieved from the reference image I_f.

First, the nearest-neighbor block is retrieved from the reference frame. Then the $K-1$ nearest neighbors of the previously retrieved block are fetched in the reference frame (cf. Figure 4.2). The matching is performed on the patch level, *i.e.* the block and its template. By using the texture information of the current block, a better prediction is obtained, as a block and its template might not be sufficiently correlated.

Only the luminance channel is considered for measuring the euclidean distance between the patches. As we are using the default 4:2:0 color sampling of HEVC, the luminance is the most significant component for the search. By performing the K-NN search in two steps, only the position of the first nearest neighbor block in the reference frame has to be transmitted on the decoder side. Since the position of the reference block is relative to the position of the current block, encoding its position comes down to encoding a motion vector, a task at which classic video coding tools are efficient.

The current template is then estimated from the retrieved templates by solving a constrained least square approximation (Equation 4.3). The vector X, composed of the pixels of the current template, is approximated as a linear combination of A, composed of the pixels of the K-NN templates, and V, the LLE weights. V is obtained by solving first $DV = 1$, with D the local covariance matrix with respect to X, then the weights are normalized to sum to one.

$$\min_V ||X - AV||_2^2 \quad \textbf{s.t.} \quad \sum_{k=1}^{K} V_k = 1 \tag{4.3}$$

From the retrieved K nearest neighbors patches, the LLE coefficients are computed from

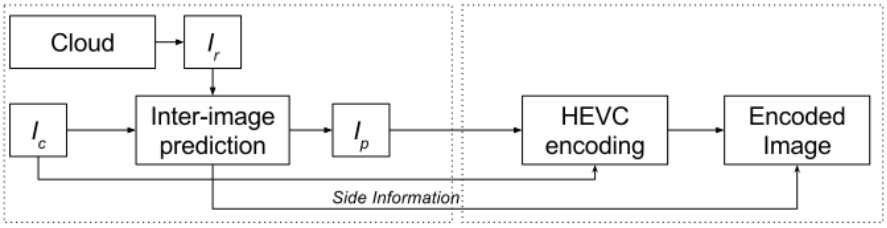


Figure 4.3: Proposed compression framework.

the templates, for each channel (Y,U,V). The current block can then be reconstructed from the retrieved blocks. By computing the coefficients from the templates instead of the blocks, there is no need to store the coefficients in the bit-stream as they are already available on the decoder side from the previously decoded blocks, which strongly increases the efficiency of the encoding. The additional cost of this local prediction amounts therefore for only one motion vector per block.

4.5 Application to image sets compression

4.5.1 Coding scheme

The coding scheme of the proposed method relies on existing video codec frameworks. HEVC was used for the experiments, but other codecs could be used as well.

The previously compensated frame is used as an extra reference frame for the compression with HEVC. To do so, the current image I_c and the predicted frame I_p are concatenated in a pseudo-video sequence and encoded as a classic video, with both intra and inter coding modes enabled. As a consequence, there is no need to implement the prediction tools in HEVC and the scheme remains agnostic to the video codec.

The side information required to reconstruct the reference frame on the decoder side is also transmitted and taken into account in the bit-rate. The side information consists of the global compensation parameters and the local motion vectors coordinates. For the decoding, the predicted image is reconstructed from the reference image and then used to decode the current image. The reference image thus needs to be available both at the encoder and the decoder sides. The reference image has to be retrieved from a large and static image database, and is referenced in the bit-stream by a single unsigned integer. The framework of the proposed compression scheme is represented in Figure 4.3.

4.5.2 Rate-distortion results

Experiments were conducted on multiple pairs of similar images presenting challenging disparities aggregated from publicly available databases [88–90], some examples are shown in Figure 4.4. These images display a combination of differences of viewpoint, focal length, illumination, rotations. The HEVC HM 16.2 software was used for the video encoding. The default parameters from the CTC *random access* mode [46] are used. For the JPEG encoding we use the ImageMagick library (6.9.0-Q16). The rate-distortion performances presented in this chapter are computed with the Bjontegaard metrics [18], using the following settings for the Quantization Parameter (QP): 22, 27, 32 and 37. The following parameters are set for the experiments: a block size of

(a) *"Building"* (b) *"Semper"*

(c) *"Market"* (d) *"Mountain"*

Figure 4.4: Examples of images contained in our dataset.

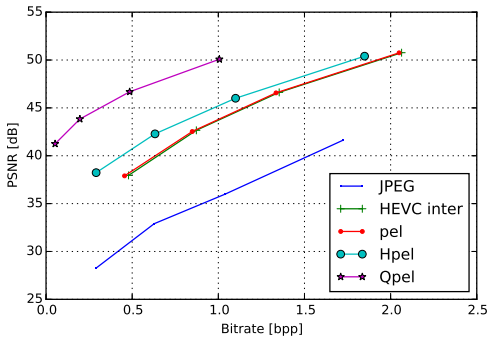
K	$p = 1$		$p = 0.5$		$p = 0.25$	
	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR
2	2.90%	-0.25db	-4.42%	0.28db	-37.61%	2.66db
4	-0.35%	0.03db	-7.91%	0.57db	-47.66%	3.45db
8	0.71%	-0.08db	-7.86%	0.58db	-49.74%	3.73db
16	0.16%	-0.01db	-10.14%	0.76db	-64.01%	5.50db
32	0.13%	-0.01db	-14.76%	1.11db	-71.66%	6.93db
64	-0.46%	0.04db	-19.90%	1.56db	-74.03%	7.50db
128	-1.52%	0.13db	-22.14%	1.76db	-75.25%	7.78db

Table 4.1: Average bit-rate savings and PSNR gains compared to HEVC inter-coding.

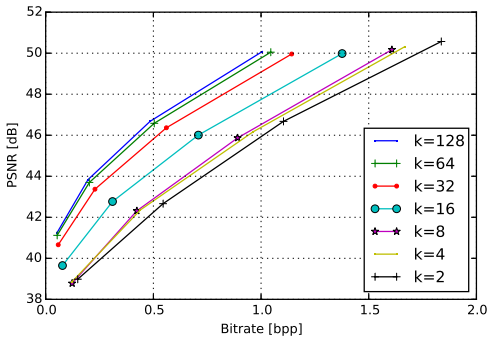
8×8 pixels and a 3 pixels wide template, composed of the upper and left pixels. Only the PSNR of the luminance channel is reported here.

Although images are sampled at the pixel level, the accuracy of block-based geometric transformations can be improved by working at sub-pixels levels. The template-matching based algorithm was thus tested with different precisions p: pixel, half-pel, and quarter-pel (*Qpel*). Also the complexity for the K-NN search being quadratic, or linearithmic at best with approximated search methods, determining the optimal number of neighbors to retrieve is critical. Several values of K ($K \in \{2^i \mid i \in [1, 7]\}$) were tested at different precisions. The results obtained for 30 pairs of images, different sub-pel accuracies and K values are presented in Table 4.1. The best performances are undeniably obtained at *Qpel* precision, where using a K value greater than 64 does not bring significant improvements (Figure. 4.5b, Figure 4.6). Also it may be observed that the order of magnitude of the gain is restricted by the (sub)-pixel precision. Different codecs and versions of our scheme are compared in Figure 4.5a.

By exploiting the inter-modes of HEVC with the predicted image, our scheme can improve the coding efficiency up to 75% bit-rate saving compared to classic HEVC inter-coding, and 97% compared to intra image coding (JPEG). While the complexity of the prediction is important,



(a) RD performance w.r.t. the precision ($k = 128$)



(b) RD performance w.r.t. K (*Qpel* precision)

Figure 4.5: Rate-distortion (RD) performance comparison.

(a) Original Image

(b) *pel* precision (21.18 dB PSNR)

(c) *half-pel* precision (24.65 dB PSNR)

(d) *Qpel* precision (29.92 dB PSNR)

Figure 4.6: Visual quality comparison ($K = 64$).

the pixel precision and the number of neighbors K can be set to constrain the processing time at the expense of the encoding performance.

4.5.3 Complexity study

The principal limitation of the LLE method is the high complexity of the algorithms involved. The main bottleneck is the K-NN search which scales in $O(DN^2)$ when performed via exhaustive matching (*i.e.* brute force matching), with D the dimension of the vector and N the number of data points. Several suggestions were proposed in [8] to reduce the complexity of the method. The complexity can be reduced by using a K-D Tree structure, a binary tree designed for efficiently organizing data points in a k-dimensional space. K-D Trees can thus be used for high-dimensional nearest-neighbors searches. The complexity of the K-NN search drops to $O(DNlog(N))$. Several implementations are publicly available, in the presented experiments the FLANN library [91] was used in the implementation.

4.6 Conclusion and perspectives

This chapter introduced a novel scheme for cloud-based image compression. Unlike previous approaches, the proposed scheme associates a global and local compensation method as an inter-

prediction method. The global compensation captures scene-wise geometric and photometric distortions, while the locally weighted template-matching prediction compensates small, local, disparities. By leveraging correlated images already stored in the cloud, the proposed method is able to encode efficiently newly uploaded images and brings significant improvements compared to classic image coding tools.

The prediction currently resides outside the HEVC encoder, an implementation in the reference HEVC HM software could be considered to take advantage of the quadtree partitioning and the advanced Rate Distortion Optimization (RDO) loop. From another perspective, the current scheme allows to use the existing coding infrastructures without introducing major modifications. The proposed method has also some limitations regarding the high complexity of the algorithms involved. Reducing the complexity of the coding scheme constitutes an important future work.

Chapter 5

Region-based models for inter-prediction

5.1 Introduction

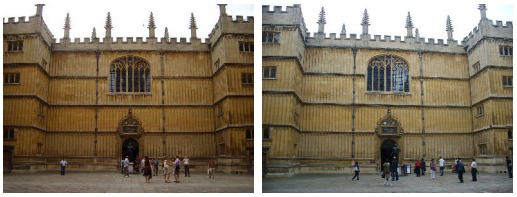
Video codecs are primarily designed assuming that rigid, block-based, two-dimensional displacements are suitable models for the motion taking place in a scene. When compressing image sets with similar images, disparities between correlated images can result from pictures taken from different viewpoints, with different cameras, focal lengths, illumination conditions, at different points in time, etc. Besides, image scenes are not always planar, and as such, multiple distortions can occur within an image pair. An example of similar images that could be found such an image set is shown in Figure 5.1. As presented in the state of the art chapter of this manuscript, several approaches have been successfully proposed to encode correlated images by compensating these distortions with multiple transformation models [21, 36]. However, in the work of Shi *et al.* [21], the number of transformation models is restricted up to 4, whereas in the method of Zhang *et al.* [36] the frame is divided into 256×256 pixel prediction units. Furthermore, both of these methods do not take into account the image content. Thus, we propose here a prediction method able to efficiently handle non-planar images with complex deformations via a region-based approach.

This chapter introduces a novel region-based prediction scheme able to leverage correlation between similar non-planar images. Region-based methods have already been demonstrated to be pertinent solutions to video compression applications [93, 94]. Unlike existing approaches, the proposed scheme extracts multiple regions, planes or objects, of the current image, each subject to a distinct transformation model. Geometric and photometric disparities are then efficiently compensated in a region-wise manner to predict the targeted frame, which is then finally encoded with HEVC [11]. As an alternative for a classical scale-offset compensation on the luminance channel, we also propose to apply a compensation model on the color channels, which is able to address larger disparities.

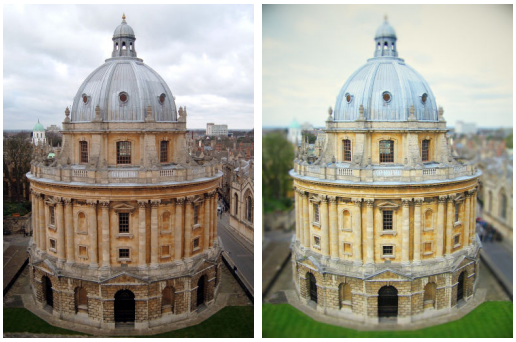
Experimental results indicate that the proposed scheme can efficiently leverage inter-image redundancy, achieving on average a -19.6% BD-rate gain compared to HEVC inter coding, computed on a dataset of several hundreds sequences. We also demonstrate that our scheme is competitive in terms of bit-rate distortion performances when compared against state of the art methods.

The rest of this chapter is organized as follows. Section 5.2 gives an overview of the proposed

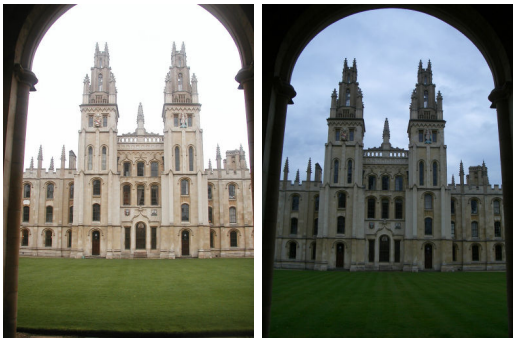
(a) "Merton College"



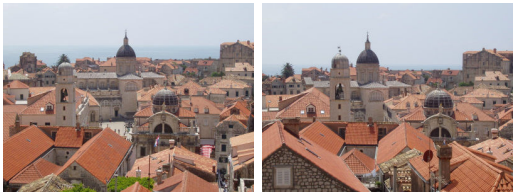
(b) "Bodleian Library"



(c) "Radcliffe Camera"



(d) "All Souls College"



(e) "Holidays 1192"

(f) "Holidays 1127"

Figure 5.1: Example of targeted images presenting geometric distortions and/or illumination disparities. Images were collected from the *Oxford Building* [92] and the *INRIA Holydays* [88] datasets. A wide variety of geometric and photometric distortions is present is these examples.

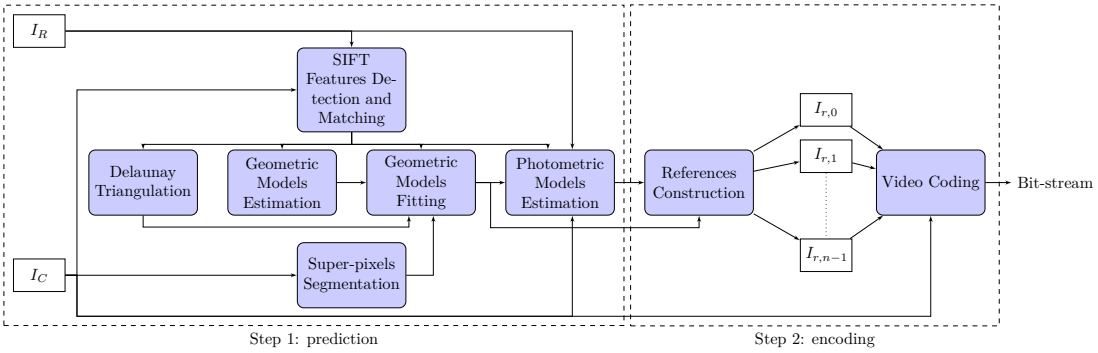


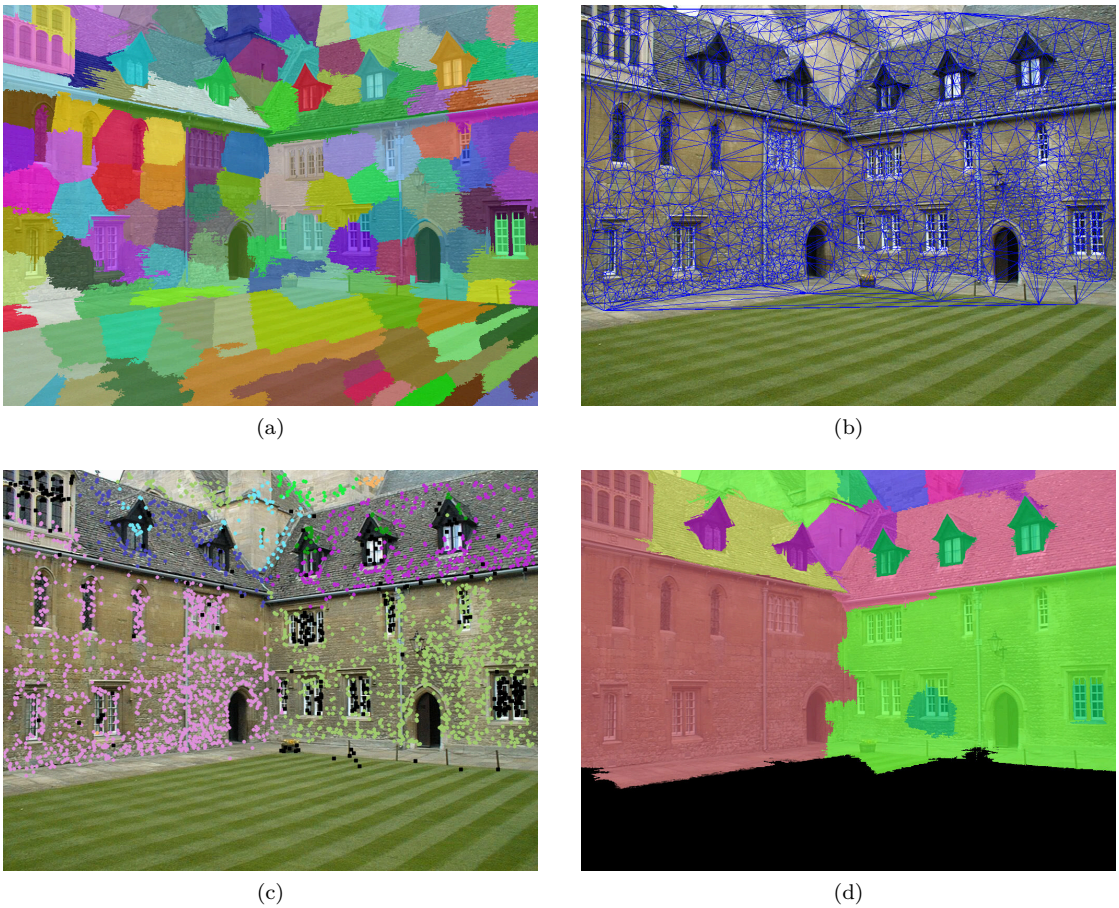
Figure 5.2: Illustration of the proposed compression scheme.

compression scheme. Section 5.3 describes it in details. Experiments on image sets and videos are reported and discussed respectively in Section 5.4 and Section 4.5.

5.2 Overview of the proposed compression scheme

The proposed compression scheme comprises two main steps, as shown in Fig. 5.2. For the purpose of explanation, we will only consider a pair of images but our scheme can also be adapted for larger sets of images. When considering the current image I_C to be encoded, a reference image I_R is first retrieved from the cloud with the help of a classical Content Based Image Retrieval (CBIR) system. Additional reference images $I_{r,i}$ are then constructed by exploiting geometric and photometric transformation models between the reference and the current images. The current image I_C is finally encoded from the reference images $I_{r,i}$ with a video encoder such as HEVC. To decode I_C, the reference images $I_{r,i}$ are reconstructed from the reference image I_R and the transformation models. The reference image thus needs to be available both at the encoder and the decoder sides. In this chapter, we assume that the reference image is retrieved from a large and static image database, and is referenced in the bit-stream.

The proposed prediction method relies on a semi-local approach which estimates region-based geometric and photometric models to better capture correlation between the two images. To segment the current image into homogeneous regions, in terms of geometric transformations, the image is first segmented into super-pixels. SIFT descriptors are then extracted from both images and matched exhaustively. For each super-pixel extracted from I_C, a projective transformation, *i.e.* a homography model, is estimated from the SIFT keypoints located inside the super-pixel boundaries. To reduce the number of homographies the estimated models are recursively re-estimated and fitted to the keypoints via the energy minimization method proposed in [95]. The Delaunay triangulation of the keypoints is used to preserve the spatial coherence during the homographies estimation. Then, the photometric disparities between I_C and I_R are compensated region-wise by estimating a transformation model between matched regions of the image pair. Multiple references $I_{r,i}$ are generated by warping each region using its assigned homographic model and applying the photometric compensation. Finally, the references are organized in a pseudo-sequence in which the current image is differentially encoded with classical video coding tools. The side information (SI), *i.e.* the homographies and the photometric model coefficients required to reconstruct the predictions on the decoder side, need to be transmitted and are taken into account in the bit-rate.



(a) (b)

(c) (d)

Figure 5.3: Region-based geometric estimation: (a) SLIC segmentation of I_C. (b) Mesh of the Delaunay triangulation of the matched keypoints. (c) Keypoints labels, each keypoint is assigned a homography model, the outliers points are represented in black. (d) Final region segmentation.

5.3 Region-based prediction

5.3.1 Super-pixel segmentation

To initialize the region-based segmentation, a super-pixel segmentation is first performed via the SLIC algorithm proposed by Achanta *et al.* in [96]. All the pixels i of I_C are clustered according to a combined colorimetric and spatial distance $D(C_k, i)$ to a centroid C_k defined as

$$D(C_k, i) = \sqrt{\left(\frac{d_c}{m_c}\right)^2 + \left(\frac{d_s}{m_s}\right)^2} \tag{5.1}$$

where d_c represents the $l2$-norm in the LAB colorspace, d_s the l_2 norm between a given pixel i and a centroid C_k. The quantities m_s and m_c are weighting parameters used to normalize color and spatial proximity. Our scheme relies on the Adaptive-SLIC (ASLIC) variant of the SLIC algorithm, where m_s and m_c are updated at each iteration of the algorithm. When using SLIC, m_s and m_c are set to constant values, the assumed maximum colorimetric and spatial distance.

Whereas with ASLIC, only the first iteration relies on fixed normalization parameters, they are then updated to the maximum distances observed in each cluster at the previous iteration. According to [96], this decreases the boundary-recall performance. However, the super-pixels compactness parameter is highly dependent on the image content and its contrast. Thus, by using the adaptive version of the algorithm, no per-image tuning is required, since the initial parameters are updated along the iterations.

The SLIC segmentation is initialized from a regular grid of centroids $\{C_k | k \in [0, K[\}$ spaced by a fixed distance, the step size s, and results in a segmentation of n super-pixels. With $K = \lfloor \frac{w}{s} \rfloor * \lfloor \frac{h}{s} \rfloor$, (w, h) the image size, and $n \leq K$ depending on the clean-up step, where some centroids with too few assignments can be removed.

An example of the resulting segmentation of the current image I_C is shown in Fig. 5.3a.

5.3.2 Geometric models estimation

To estimate the geometric models, our scheme relies on local feature descriptors as they are more robust to geometric distortion (*e.g.* translation, rotation, zoom, scale) and illumination variations than the pixel values [23].

SIFT keypoints are first extracted from both I_C and I_R and then matched exhaustively. In order to improve the matching, we use the RootSIFT algorithm proposed by Arandjelovic *et al.* in [97]. The computed SIFT descriptors X_i are first projected into a feature space:

$$X_i^{'} = \sqrt{\frac{X_i}{\|X_i\|_1}}, \forall i \in [\![1, N]\!] \tag{5.2}$$
$$with \ \ \|X_i\|_1 = \sum_{j=1}^{128} |X_i(j)|$$

then the distance between them is computed using the l_2 norm. For each super-pixel, a homography model H, defined by the matrix

$$H = \begin{bmatrix} s_x.\cos(\theta) & -s_y.\sin(\theta + \sigma) & t_x \\ s_x.\sin(\theta) & s_y.\cos(\theta + \sigma) & t_y \\ k_x & k_y & 1 \end{bmatrix} \tag{5.3}$$

is then estimated via the RANSAC [22] algorithm from the matched keypoints contained within the super-pixel boundaries. Here (t_x, t_y) denote the translation coefficients, θ the rotation, (s_x, s_y) the scale parameters, σ the shear, and (k_x, k_y) the keystone distortion coefficients.

RANSAC is an iterative method which can estimate a parametric model from a noisy set of data points. There is no guarantee that the optimal solution will be found during the iterations. However, the probability of success is independent of the number of points in the data set and only relies on two parameters: the number of iterations N and the residual threshold t to discard an outlier. Let u be the probability of a data point to be an outlier, the minimal number of iterations to reach a probability p of finding the optimal solution is given by

$$N = \frac{\log(1 - p)}{\log(1 - u^c)} \tag{5.4}$$

where c is the minimum number of samples to estimate the parametric model. In the case of a homography model, $c = 4$ (8 degrees of freedom).

To robustly estimate a homography model with RANSAC, the Symmetric Transfer Error (STE) [98] is used to compute the distances between matched keypoints:

$$STE(H_l) = \overbrace{\sum_{p \in P} d(x'_p, H_l.x_p)^2}^{\text{forward term}} + \overbrace{\sum_{p \in P} d(x_p, H_l^{-1}.x'_p)^2}^{\text{backward term}} \tag{5.5}$$

where H_l denotes a homography model to be evaluated, x_p and x'_p two matched keypoints, P the set of matched keypoints, and d the euclidean distance. Since the STE takes into account both forward and backward projections of matched keypoints, this distance is well suited for real-world data where local feature detection and their matching will likely contain errors [98].

To further improve the estimation process, the determinant of the homography matrix is also used to discard invalid models. As pointed out by Vincent *et al.* in [99], homographies not respecting the condition:

$$\mathcal{H} = \left\{ H_l \mid \frac{1}{k} \leq |\det(H_l)| \leq k \right\} \tag{5.6}$$

can be rejected as they correspond to degenerated cases, *i.e.* the absolute value of the determinant of the matrix (or its inverse) is close to zero. Following the recommendation of [99], we set k to 10.

From the n super-pixels of the SLIC segmentation, m homography models are thus estimated, with $m \leq n$. Indeed, some super-pixels do not contain a sufficient number of matched keypoints to estimate a projective transform, or contain only outliers. Furthermore, the models attributed to neighboring super-pixels may be highly similar as they might be part of the same region.

5.3.3 Geometric models fitting

From the previously estimated homography models, the most representative model for each region needs to be extracted and refined before generating the projections.

Delong *et al.* proposed in [95] an efficient method to solve the issue of multiple models fitting. To solve this labelling problem, *i.e.* assigning a model to each keypoint, they introduce a new joint discrete energy:

$$E(f) = \overbrace{\sum_{p \in P} D_p(f_p)}^{\text{data cost}} + \overbrace{\sum_{(p,q) \in N} V_{pq}(f_p, f_q)}^{\text{smooth cost}} + \overbrace{\sum_{L \subseteq \mathcal{L}} h_L.\delta_L(f)}^{\text{label cost}} \tag{5.7}$$

to be minimized iteratively, where N is the keypoints neighborhood, h_L the label cost of the subset of labels L, and where the function $\delta_L(f)$ is defined as:

$$\delta_L(f) \triangleq \begin{cases} 1, & \exists p : f_p \in L \\ 0, & \text{otherwise} \end{cases} \tag{5.8}$$

Following the set-up described in [100], an initial proposal for the homography models needs to be estimated from the matched keypoints, the observations P. During the expansion step, each keypoint p is assigned a label l from the set of homographies L in order to minimize the objective function (5.7). From the labelling f, the set of models can then be updated (re-estimation step). The expansion and re-estimation steps are performed iteratively until convergence of the minimization of (5.7) or until a maximum number of iterations is reached.

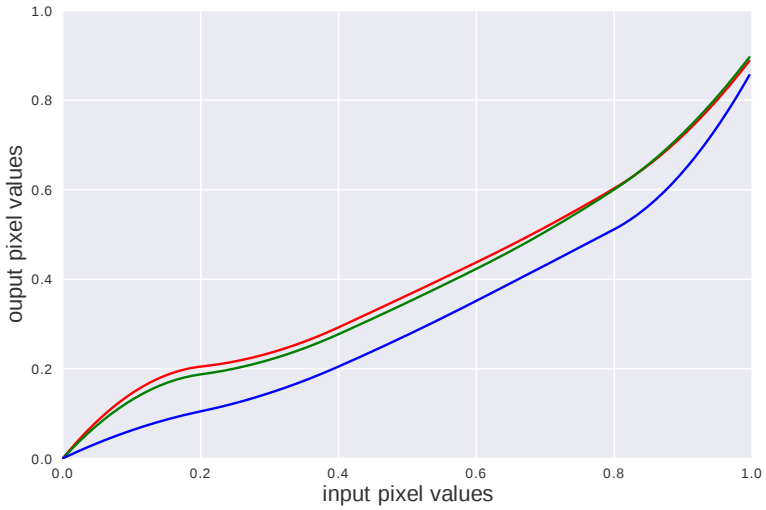


Figure 5.4: Example of the splines fitting on the RGB channels.

In the set-up described in [95] and [100], the set of initial homography models is randomly generated by selecting N samples of 4 matches. In our approach, we use the models previously estimated from the super-pixels, which allows a faster convergence and a more robust estimation. The set of homography models is then reduced and refined by recursively minimizing the energy (5.7).

The data cost is a fidelity term, which ensures that the model properly describes a transformation, computed from the STE (5.5). Due to the likely presence of outliers in the matches, an additional model ϕ is introduced to fit their distribution, with a fixed data cost for all the vertices and a label cost set to zero:

$$\begin{cases} h_\phi = 0 \\ D_p(\phi) = C, \text{with } C > 0 \end{cases} \tag{5.9}$$

The smoothness cost for the set of neighbors $(p, q) \in N$ is defined from the Delaunay triangulation of the matched keypoints in the current image I_C (Fig. 5.3b). It penalizes neighboring points with different labels in order to preserve spatial coherence and is defined as:

$$V_{pq} = w_{pq} \cdot \delta(f_p \neq f_q) \text{ with } \begin{cases} w_{pq}, & \text{weight for the vertex } pq \\ \delta, & \text{Kronecker delta} \end{cases} \tag{5.10}$$

The label cost (5.8) is used to restrict the number of models.

An example of the resulting labelling is shown in Fig. 5.3c, where one can observe that several planes, or regions, of the image are detected successfully.

5.3.4 Photometric compensation

Once the finite set of homographies describing geometric transformations between image pairs has been determined, a reference image can be constructed. However, disparities due to illumination

and photometric differences between the constructed reference image and the current image persist. During the encoding, these disparities will result in a highly energetic residual, limiting the use of the predicted image by the encoder.

To compensate these distortions, we propose to estimate a photometric compensation model for each previously estimated region.

A scale-offset model is often proposed to minimize distortion on the Y channel ([21, 27, 36]). The model coefficients, α and β are computed by minimizing the sum of square errors on the matched keypoint pixels:

$$\underset{\alpha,\beta}{\operatorname{argmin}} \sum_{P} |Y'(x'_p) - (\alpha.Y(x_p) + \beta)|^2 \tag{5.11}$$

This model can efficiently handle illumination disparities, but performs poorly on complex colorimetric disparities. We choose to add the more flexible model proposed by Hacohen *et al.* in [101]. The photometric deformation is modelled by a piece-wise cubic spline f on each RGB channel. This model can compensate for a variety of photometric distortions such as gamma changes or color temperature. The minimization problem:

$$\begin{aligned} \underset{f}{\operatorname{argmin}} \quad & \sum_{Q} |I'(x'_q) - f(I(x_q))|^2 + C_{soft}(f) \\ \text{subject to:} \quad & C_{hard}(f) \end{aligned} \tag{5.12}$$

is solved for 6 knots (0, 0.2, 0.4, 0.6, 0.8, 1) via quadratic programming. The same soft constraints (C_{soft}) and hard inequalities constraints (C_{hard}):

$$\begin{aligned} C_{soft}(f) = \lambda_1 & \sum_{x \in \{0,1\}} |f(x) - x|^2 \\ + \lambda_2 & \sum_{x \in \{0.2j-0.1\}_{j=1}^{5}} |f(x) - x|^2 \\ + \lambda_3 & \sum_{x \in \{0.2j-0.1\}_{j=1}^{5}} |f''(x)|^2 \end{aligned} \tag{5.13}$$

$$C_{hard}(f) = \begin{cases} 0.2 \leq f'(x) \leq 5, \forall x \in \{0.2j - 0.1\}_{j=1}^{5} \\ f(0) \leq 0 \end{cases} \tag{5.14}$$

are used to control smoothness and monotonicity of the curves. Hard equality constraints are also set on the 4 inner knots of the splines and their first derivative. Each curve thus has 7 degrees of freedom.

The minimization is performed for each region determined from the labelling. As we cannot rely on a dense correspondence field as in the original paper [101], we use a set of pixels Q within a given radius of matched keypoints of each region, to ensure that only reliable pairs of pixels values are used.

We use the sum of absolute differences (SAD) to select the best performing photometric model for each region during the prediction. The SAD is preferred here over the sum of squared differences (SSD), as it tends to favour more compact residuals, and thus is considered as a better estimator for the quality of the reconstruction. The photometric compensation can also be disabled when the image pair does not present any photometric distortions or the estimation fails.

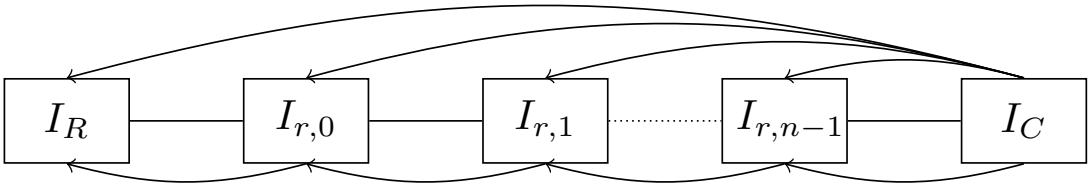


Figure 5.5: Illustration of the pseudo-video sequence encoding scheme.

5.4 Application to image set compression

5.4.1 Coding scheme

Once the geometric and photometric models have been successfully estimated, the image can be segmented into regions at the pixel level. The region segmentation is computed by selecting the best projection for each super-pixel. The mean absolute error is used to measure the distortion for each super-pixel between a given projection and the current image. An example of the final segmentation is shown in Fig. 5.3d.

A prediction image can then be constructed from the reference frame, the estimated models and the region segmentation. However, sending this segmentation map to reconstruct the prediction on the decoder side would be costly. Instead, multiple reference pictures are used, which can be constructed in the same manner on both the encoder and decoder sides.

Those n additional references $I_{r,i}$ are constructed from the reference image I_R and the region models (the associated geometric and photometric compensation models). For each region, an additional reference image $I_{r,i}$ is constructed by warping the reference image I_R with the associated region homography model and applying the photometric correction. This step is performed both at the encoder and decoder side, and as such the encoded $I_{r,i}$ are discarded in the transmitted bit-stream. The reference image, the projections and the current image are then concatenated in a pseudo-video sequence, finally encoded with HEVC. Our encoding structure differs from the main HEVC profiles such as the low-delay and hierarchical configurations, as the last frame needs to be predicted from all the previous frames in the sequence in order to fully exploit the inter-redundancies.

Starting from the low-delay configuration of the HM software (*lowdelay_P_main.cfg*), the GOP settings are modified to keep all the frames in the reference pictures buffer, as shown in Fig. 5.5. The reference frames are encoded at maximum quality ($QP = 0$), since this part of the bit-stream will not be stored in the final bit-stream, while the quality of the current frame is controlled via the QP_{offset} value.

To enable the decoder to reconstruct the projections used as reference pictures for the current image, some Side Information (SI) is also stored alongside the HEVC bit-stream. By using several reference frames, only the geometric and photometric models coefficients need to be transmitted. The encoder then performs its reference selection for each inter-coded prediction unit and stores it in the bit-stream. This avoids sending the costly segmentation map, and lets the encoder decide the best reference frame to select for each prediction-unit, in the rate distortion optimization (RDO) loop. All the SI parameters are stored as half-precision floating point, coded on 16 bits each. For the homography models, 8 parameters need to be stored in the bit-stream, 2 parameters for the scale-offset model or 7 parameters for each color channel for the piece-wise spline fitting model, as detailed in Tab. 5.1.

Table 5.1: Side information (SI) sent to the decoder for each of the n predicted regions. For the photometric compensation, one of the two models is chosen for each region, or the compensation is disabled.

Compensation method	**Model**	**Bits**
Geometric	*homography*	$8 * 16$
Photometric	*scale-offset*	$2 * 16$
	piece-wise spline	$7 * 16$

5.4.2 Rate-distortion results

To perform our experiments, numerous images have been retrieved from publicly available databases [88–90, 102] and also crawled from Google Images and Flickr. The collected images present challenging disparities such as combinations of different viewpoints, focal lengths, illumination variations, translations, rotations. Such disparities result from pictures taken at different points in time, with different camera positions, lighting conditions, etc...

Unless otherwise specified, the HEVC HM[1] software version 16.9 with the *low-delay* configuration is used for the video coding in all the following tests. The rate-distortion performances presented in the rest of this chapter are computed with the Bjontegaard metric [18] using the recommended settings of 22, 27, 32 and 37 for the Quantization Parameter (QP). The PSNR is computed on the Y channel.

The super-pixel centroids are initialized on a regular grid, spaced by 64 pixels. The initial compactness is set to 10. In order to use the energy (5.7) to estimate the multiple geometric models, the value of the label, smooth and outlier costs first need to be determined. As stated by Delong *et al.* in [95], these parameters are application dependent, and as such, they can be learned offline once, on a representative dataset. Their values have been computed on a training dataset with the differential evolution algorithm introduced by Storn & Price in [103]. This method allows finding the global minimum of a multivariate function, over a large space of possible parameters combinations more robustly than with manual tuning, to the detriment of a slow convergence. Based on empirical evaluations, the super-pixels size is set to 64 pixels.

As regards the splines fitting based photometric model, the parameters provided in [101] have been used. The pixel search radius is set to 15 pixels and the quadratic problem has been solved with an efficient quadratic solver[2].

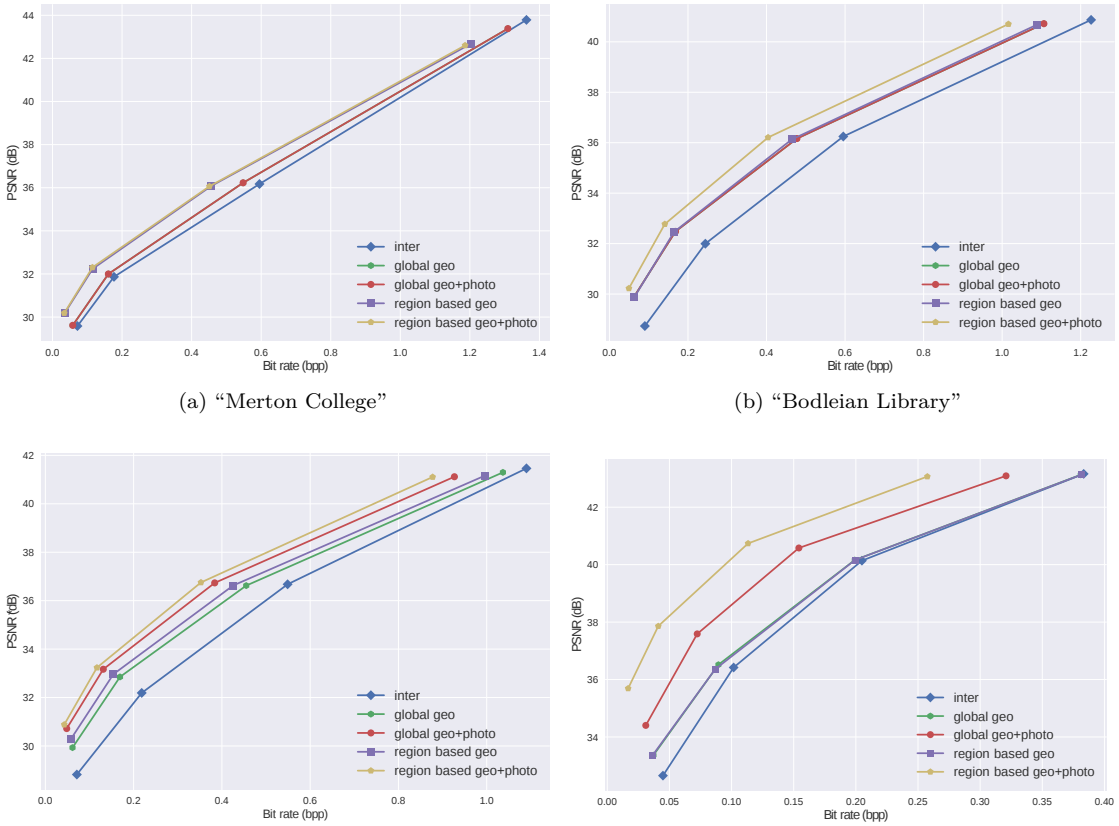
The same set of parameters is used for all the results presented in this chapter.

Performance of region-based models

The performance of the region-based prediction model ("region-based") is first compared with the performance of a global compensation model ("global") and also with HEVC low-delay ("inter"). The two prediction modes are evaluated with only the geometric compensation enabled ("geo") and both the geometric and photometric compensations enabled ("geo+photo"). The global compensation scheme consists in a single homography transformation, estimated from the classical SIFT+RANSAC approach, followed by a photometric scale-offset compensation, *i.e.* with only one homography and one photometric compensation model per image. Examples are shown in Fig. 5.6 for the four sequences of Fig. 5.1: "Merton College" (Fig. 5.6a), "Bodleian Library" (Fig. 5.6b), "Radcliffe Camera" (Fig. 5.6c), and "All Souls College" (Fig. 5.6d), the

[1] `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/`

[2] Available online: `https://github.com/liuq/QuadProgpp`



(a) “Merton College”

(b) “Bodleian Library”

(c) “Radcliffe Camera”

(d) “All Souls College”

Figure 5.6: Performance comparison of the different prediction methods. The proposed region-based model and a global scheme are compared to HEVC inter, with and without photometric compensation.

Bjontegaard metrics are provided in Tab. 5.2. The “Merton College” sequence exhibits multiple geometric distortions, while “Bodleian Library” and “Radcliffe Camera” both present multiple geometric and photometric distortions, and “All Souls College” a global geometric and photometric distortion. The choice of which image is the current/reference image was performed randomly.

Table 5.2: Bjontegaard metrics computed on the rate-distortion curves presented in Fig. 5.6. The symbol "r" indicates that the sequence was processed backward, *i.e.* the reference image and the current image were switched.

Sequence	**global geo**		**global geo+photo**		**region-based geo**		**region-based geo+photo**	
	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR
"Merton College"	-9.96%	0.42db	-9.96%	0.42db	**-28.50%**	**1.22db**	**-28.50%**	**1.22db**
"Bodleian Library"	-26.94%	1.40db	-26.94%	1.40db	-28.03%	1.47db	**-39.80%**	**2.17db**
"Radcliffe Camera"	-17.78%	0.90db	-34.18%	1.84db	-26.03%	1.34db	**-40.54%**	**2.26db**
"All Souls College"	-10.63%	0.51db	-38.63%	2.18db	-10.55%	0.50db	**-61.32%**	**3.79db**
"Merton College" r	-15.52%	0.74db	-15.52%	0.74db	**-28.32%**	**1.39db**	**-28.32%**	**1.39db**
"Bodleian Library" r	-36.65%	1.98db	-46.19%	2.65db	-37.61%	2.04db	**-53.45%**	**3.21db**
"Radcliffe Camera" r	-19.95%	0.93db	-32.64%	1.64db	-25.02%	1.21db	**-38.66%**	**2.01db**
"All Souls College" r	-11.15%	0.50db	-23.30%	1.12db	-11.80%	0.53db	**-38.16%**	**1.88db**

The proposed scheme results in the following respective BD-rate improvements of -28.50%, -39.80%, -61.32% and -40.54% compared with HEVC inter. For the "Merton College" sequence, the BD-rate gain increases from -9.96% to -28.50% thanks to the use of multiple geometric compensation models, whereas the photometric compensation does not yield any performance improvements for this sequence.

On the "All Souls College" sequence, one might observe that the photometric compensation can greatly improve the efficiency, from -10.63% to -38.63%. Also, in this case, the photometric compensation of the region-based model is more performant, from -38.63% to -61.32%. Although there is only one region in the image, the photometric model based on splines yields a better prediction. This is confirmed on the "Bodleian Library" and "Radcliffe Camera", which all benefit from the photometric compensation, from -28.03% to to -39.80% and from -26.03% to -40.54%, respectively. It is also worth noting that on the "Radcliffe Camera" and "Bodleian Library" sequences, the global scheme with the photometric compensation performs better than the region-based algorithm without photometric compensation, emphasizing its crucial role in providing an accurate prediction.

For the four sequences, the respective bit-stream ratio allocated for the side -information over the total bit-stream size is -0.77%, -0.41%, -0.28% and -0.25%, which is negligible.

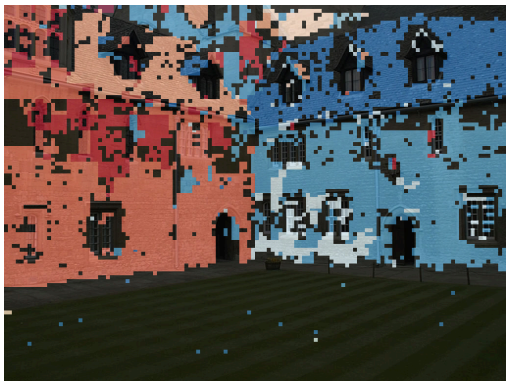
Results for the reversed sequences, where the reference and current images are swapped, are also provided for comparison. The rate-distortion improvements are consistent for the "Merton College" and "Radcliffe Camera" sequences. However, while an increase in performance is observed for the "Bodleian Library" sequence, one can notice a decrease for the "All Souls College". This can be explained by the different exposures of the frames in each sequence. Indeed, predicting an image from an under-exposed correlated image is more challenging, as numerous details are lost due to the lack of brightness and thus cannot be predicted correctly.

To illustrate the performance gains due to the use of the predicted regions, we modified an HEVC bit-stream analyzer to display the reference picture index used for each coding unit. An example is shown Fig. 5.7 on the "Merton College" sequence, where the encoder decisions for the reference picture selection are displayed for different QP values. Each color indicates a reference frame, *i.e.* a region, chosen by the encoder as a reference picture, the intra mode is represented in black. One can observe that the reference frame selection for each coding unit in the quad-tree is overall quite consistent with the region-based segmentation presented previously. Still, there are some local inconsistencies in the reference selection that can be attributed to the decisions of the RDO loop. Also, at higher bit-rates, *i.e.* lower QP values, the intra-mode is more frequently selected by the encoder, especially in complex zones such as the windows where the light reflection cannot be predicted accurately.

Comparison to the state of the art

To compare with the state of the art, we implemented the approaches proposed by Shi *et al.* [21] and Zhang *et al.* [36]. Both methods were re-implemented based on their respective publications. The BD-rate gains are reported in Tab. 5.3 for the six sequences shown in Fig. 5.1. Our method achieves a higher coding performance for the image pairs (a), (b), (d), (e) and (f), with a respective improvement of -3.9%, -12.28%, -16.62%, -3.91% and -3.45%.

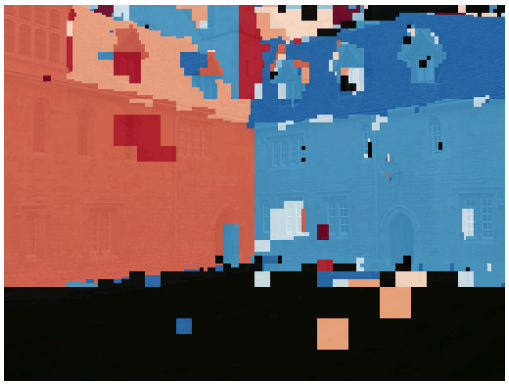
Improvements over the state of the art can be explained by the use of a finer prediction model. Restricting the number of models to 4, as proposed by Shi *et al.*, reduces the prediction efficiency as smaller regions could be absorbed into larger ones and thus would not benefit from an accurate prediction model. Zhang *et al.* divide the images into 256x256 pixel "units", which is costly in terms of side information which needs to be encoded, this explains the lower performance on all the sequences compared to our scheme. A "unit" can also span over multiple



(a) $QP = 22$

(b) $QP = 27$

(c) $QP = 32$



(d) $QP = 37$

Figure 5.7: Encoder reference frame decisions by coding unit on “Merton College” for different QP values. Each color corresponds to a reference frame index in the active reference picture set (the projections for our application), while the intra-mode is represented in black.

Table 5.3: BD-rate reduction compared with HEVC inter for different methods, with N the number of prediction models.

Sequence	**Zhang [36]**		**Shi [21]**		**Ours**	
	BD-rate	N	BD-rate	N	BD-rate	N
"Merton College"	-19.63%	12	-24.60%	4	**-28.50%**	8
"Bodleian Library"	-19.17%	12	-27.52%	4	**-39.80%**	6
"Radcliffe Camera"	-28.82%	12	**-42.74%**	4	-40.54%	3
"All Souls College"	-26.42%	12	-44.70%	4	**-61.32%**	1
"Holidays-1192"	-4.59%	80	-1.67%	4	**-8.04%**	7
"Holidays-1127"	-23.94%	80	-33.29%	4	**-37.2%**	4
"Merton College" r	-13.29%	12	-21.17%	4	**-28.32%**	8
"Bodleian Library" r	-19.80%	12	-41.96%	4	**-53.45%**	6
"Radcliffe Camera" r	-31.02%	12	-37.37%	4	**-38.66%**	3
"All Souls College" r	-31.55%	12	-36.50%	4	**-38.16%**	1
"Holidays-1192" r	-3.8%	80	-7.16%	4	**-12.59%**	8
"Holidays-1127" r	-25.41%	80	-27.95%	4	**-32.07%**	4
Mean BD-rate gain	-20.64%		-28.89%		**-34.89%**	

planes and thus results in an incorrect projection estimation. In our scheme, the regions are dependent on the image pair content correlations, thus the models are more robustly estimated from a plain and distinct region, which results in a better prediction. Moreover, the proposed photometric compensation on the color channels can be more efficient than the simple scale-offset compensation model on the luminance channel. For the (c) sequence, our scheme fails to detect the optimal number of models and selects 3 models instead of 4. With a fixed number of 4 models, an improvement of -44.56% over HEVC inter can be obtained. As such, the automatic detection of the number of regions performs well on average, but can result in lower gains on some sequences. Re-learning the parameters of the model fitting on a larger dataset could help improving the performances.

Overall performance

In this section we present the overall coding performance of the proposed prediction scheme on a large dataset of image pairs. Multiples images were randomly aggregated to form a collection of about 700 sequences from the previously mentioned online databases [88–90, 102]. This dataset presents a large variety of scene contents, image resolutions and distortions: different cameras, viewpoints, conditions of illuminations, etc... Again, the BD-rate gains reported here are calculated with respect to the performance of the HEVC low-delay inter-coding configuration.

The overall performances in terms of BD-rate saving for different geometric prediction methods, a single model "global" versus our method "region-based", with and without photometric compensation, are shown in Fig. 5.8.

The mean BD-rate distortion improvements are respectively -12.16%, -13.29%, -16.50% and -19.62%. The distribution of the rate-distortion improvements on the dataset is shown in Fig. 5.9. The wide range of gain interval (from -0% to -61%) reflects the method high dependency on the inter-image correlation. As such the expected gain is highly dependent on the image pair content. The full region-based model outperforms the other models, with at least -19.62% gain for half the sequences. While 41.59% of the sequences do not benefit from a photometric compensation,

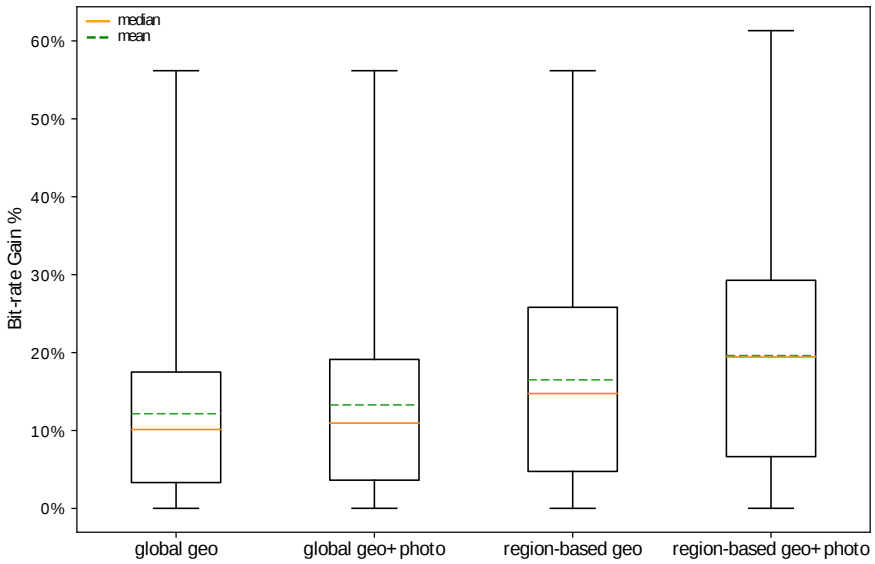


Figure 5.8: Overall performance comparison of prediction methods.

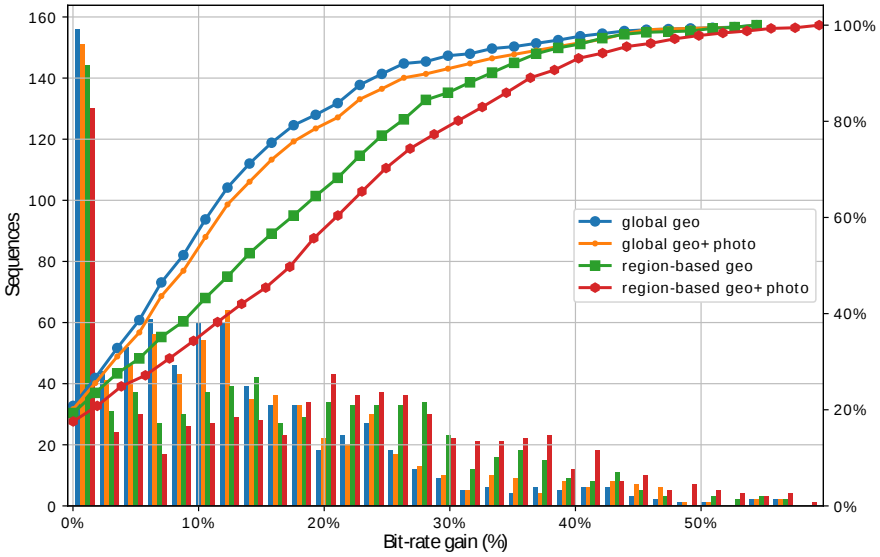


Figure 5.9: Distribution of the rate-distortion gains for different prediction methods, with their respective cumulative density function. Lower is better, as the cumulative density increases more slowly due to higher gains. The greater performance of the region-based model with photometric compensation can be clearly noticed.

Table 5.4: Mean runtime increases for the total encoding and the HEVC encoding of the proposed scheme, compared to the HEVC inter-coding of two images.

Method	**Total**	**HEVC encoding**
global geo	121.54%	110.59%
global geo+photo	120.61%	109.65%
region-based geo	143.17%	125.76%
region-based geo+photo	142.55%	123.45%

Table 5.5: Distribution of the runtime for each step in the region-based scheme.

Step	**Runtime ratio**
Super-pixels extraction	26.58%
Descriptors extraction and matching	44.86%
Geometric models estimation	2.64%
Geometric models fitting	24.10%
Photometric compensation	1.46%
Misc.	0.36%

the scale-offset model is selected for 37.61% of the sequences and the piece-wise spline model is more performant on the remaining 20.80%. The higher gains are obtained for frames with a simple global geometric distortion, such as a rotation, which cannot be compensated efficiently by the block motion estimation and compensation of video encoders. The low gains result from either sequences with complex distortions that cannot be compensated with geometric-based compensations (*e.g.* significant optical distortion) or simple distortions already compensated efficiently by block motion compensation. Also our scheme strongly relies on the keypoints extraction and matching step, which can fail for some scenes, as no adaptive method is proposed to control the sensitivity of the detector and the matching threshold. In these cases, no prediction can be performed and thus no improvements over the HEVC "inter" baseline can be expected.

For the BD-rate distortion improvement over HEVC all-intra coding, a mean gain of -21.56% is achieved. The gain of -19.62% obtained over HEVC "inter" indicates that the HEVC inter-prediction models can only handle larger distortions to a limited extent.

5.4.3 Complexity study

Compared to a classical pseudo-video coding approach, the main increase in complexity of our scheme resides on the encoder side. The mean encoding run-times of HEVC low-delay, a global prediction scheme and our region-based prediction model have been computed for the same 700 sequences, and are reported in Tab. 5.4. The mean runtime increase of our scheme is of 142.55% compared with HEVC inter. The increased complexity can be explained both by the overhead of the region-based prediction algorithm and the HEVC inter-coding.

The distribution of the increase in complexity of the region-based prediction algorithm is detailed for each step in Tab. 5.5. The slowest step is by far the local descriptors extraction and matching, followed by the super-pixels extraction and the homography models fitting. The SIFT extraction could benefit from a more efficient implementation, such as the GPU one proposed in [104]. The descriptors matching could also be performed on GPU, or leverage the approximate k-nearest-neighbors methods based on KD-Trees such a FLANN [91]. Similarly, an efficient GPU

Table 5.6: Influence of the "search range" value on the runtime and the rate-distortion performance.

Prediction method	**Search Range**	**Runtime**	**BD-rate gain**
"global"	64px	116.45%	14.44%
"region-based geo"	1px	105.90%	14.98%
	2px	107.18%	15.31%
	4px	107.24%	15.50%
	8px	107.57%	15.69%
	16px	108.76%	15.72%
	32px	111.14%	15.67%
	64px	130.73%	16.50%

implementation of SLIC has also been proposed [105].

The second significant overhead in the complexity of the proposed scheme is due to the inter-prediction process in HEVC. The 23.45% increase is due to the compensated regions which need to be encoded by HEVC before being available as reference to encode the target frame.

By enforcing multiple reference frames, the encoder has to perform more block motion estimations to compute the potential motion vectors. The default "low-delay" configuration of HEVC sets a search range of 64 pixels for the motion vector, and can be reduced to speed up the encoding at the cost of a reduction of the compression performance. Experimental results are reported in Tab. 5.6 for the 700 sequences. One can observe that by setting a lower value for the motion vector search range, the encoding runtime can be reduced, at the expense of a decreased BD-rate gain, since local geometric disparities would not be well compensated by the constrained block motion compensation. However, it provides a good trade-off between complexity and efficiency.

On the decoder side, the increase in complexity is fairly limited. Once the reference image has been retrieved, only the additional step of reconstructing the projections from the side information needs to be performed. This step amounts to computing the new pixel coordinates, applying an interpolation and finally correcting the pixel values with the photometric model for each region. This operation has a $O(n)$ complexity, and as such, can be performed in linear time with respect to the image size. With our implementation, it takes less than 1s to generate the reference images on a recent laptop.

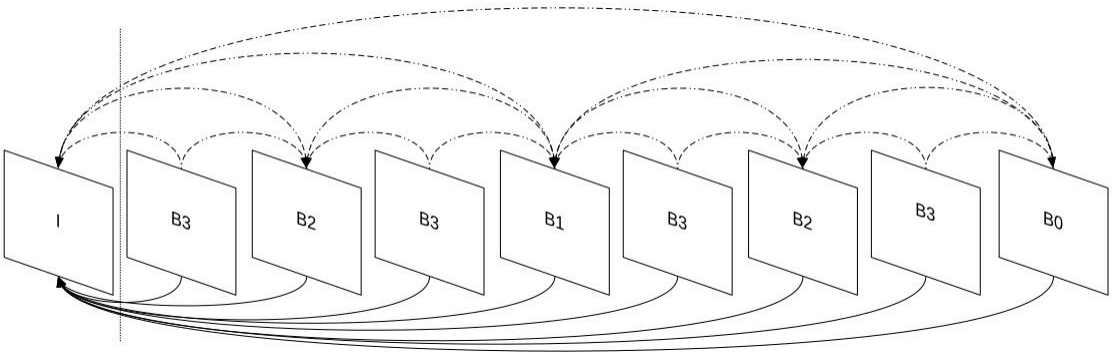


Figure 5.10: Coding scheme: illustration for a GOP of 8 frames with the default *random access* configuration. The dashed lines on top represent the default references, the straight lines below the extra reference for the proposed mode.

5.5 Application to video compression

The previously described semi-local prediction method is tested here in a video compression context. The prediction scheme was adapted and implemented into the HM software as an extra inter-prediction mode. The HEVC bit-stream was modified accordingly to signal the use of this additional prediction method.

5.5.1 Coding scheme

When compressing a video sequence with a classical video codec, most of the sequence frames are coded with the inter-prediction mode enabled, to make use of the temporal redundancy. All frames coded with inter-prediction leverage a set of reference frames, from which block predictions are performed by estimating and transmitting motion vectors and residues in the bit-stream. The proposed region-based estimator was implemented in this context.

In our setup the region-based models are estimated between the original current frame and the original first frame of the group of picture (GOP). The coding scheme is illustrated in Figure 5.10. Although the predictor could be used for all the frames in the reference pictures buffer of the current image, we choose to use only one frame for implementation reasons, focusing on demonstrating the efficiency of the proposed mode. Potential prediction blocks are generated for each extracted model by warping and interpolating the reconstructed (encoded/decoded) blocks of the reference frame from the homography model, then compensating the luminance channel. The estimation process is depicted in Figure 5.11.

Once all the blocks of the current frame have been encoded, we determine which models have been actually used by the encoder through the rate distortion optimization (RDO) process. The default inter-prediction modes of HEVC are often efficient enough to predict the blocks and sending the parameters for multiple models is more expensive than simple motion vector parameters. As such, our mode competes with the other inter-prediction modes in the RDO loop and is only activated when the classical translational estimation fails to predict correctly the current block.

A specific syntax is added in the HEVC bit-stream to signal the used models, so that the stream can be decoded.

A flag is first set at the frame level to signal the use or not of at least one of the extra reference frames. At the Prediction Unit (PU) level, an integer is added to the motion vector information

Table 5.7: BD-rate reduction compared with the HEVC baseline. Results for the affine sequences are reported in set (1), and non-affine sequences in set (2). "GLB" refers to the global compensation scheme, "RB" to the region-based one. The "-L" suffix indicates the use of the luminance compensation method.

Sequence		GLB	GLB-L	RB	RB-L
(1)	CStoreGoods_720x1280	-2.29%	-2.29%	-2.58%	**-2.71%**
	DrivingRecorder1_720x960	0.16%	0.16%	-0.90%	**-1.05%**
	DrivingRecorder2_720x960	0.11%	0.07%	-1.63%	**-1.81%**
	LakeWalking_720x960	-7.06%	-7.06%	-9.26%	**-10.45%**
	ParkSunny_720x1280	-0.13%	-0.37%	-0.28%	**-0.74%**
	ParkWalking_720x1280	-2.04%	-2.04%	-2.73%	**-3.11%**
	bluesky_1920x1080	-3.90%	-4.21%	-3.97%	**-4.32%**
	station2_1920x1080	-11.28%	-11.75%	-12.13%	**-13.27%**
	tractor_1920x1080	0.53%	0.16%	**-0.27%**	**-0.27%**
	average	-2.88%	-3.04%	-3.75%	**-4.19%**
(2)	B_Cactus_1920x1080	0.02%	-0.03%	-3.16%	**-3.17%**
	city_1280x720	0.13%	0.13%	0.00%	**-0.26%**
	in_to_tree_1280x720	-0.28%	-0.28%	**-0.54%**	**-0.54%**
	shields_1280x720	-2.06%	-2.06%	**-2.13%**	**-2.13%**
	average	-0.55%	-0.56%	-1.46%	**-1.52%**
Total average		-2.16%	-2.27%	-3.04%	**-3.37%**

to store the index of the extra reference frame used. The sentinel value 0 is set as a signaling method to inform the decoder that no extra reference frames are used for the current PU. The interpolated frame index is entropy coded with CABAC in order to reduce the syntax size. A simple CABAC context of one bit is used. The geometric and photometric models parameters are also encoded and stored in the bitstream for each frame, as half-precision floating point (16 bits).

5.5.2 Experimental results

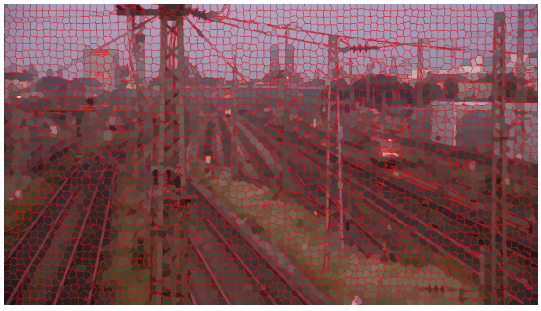
The coding experiments are performed on common test sequences [46, 106] and proposed User Generated Content (UGC) sequences [107]. The selected sequences display a wide variety of motion caused by camera zooms, camera rotations, camera shakes, and classical 2D translational motion.

The HEVC HM software version 16.16 was used for all the experiments. The rate-distortion performances presented here are computed with the Bjontegaard metric [18] using the common 22, 27, 32, 37 Quantization Parameter (QP) values. The PSNR is reported on the Y channel only. The default HM *random access* configuration mode [46] is used as a baseline in all the following tests, with the default fixed GOP size of 16. The parameters for the region-based models estimator are fixed for all the experiments.

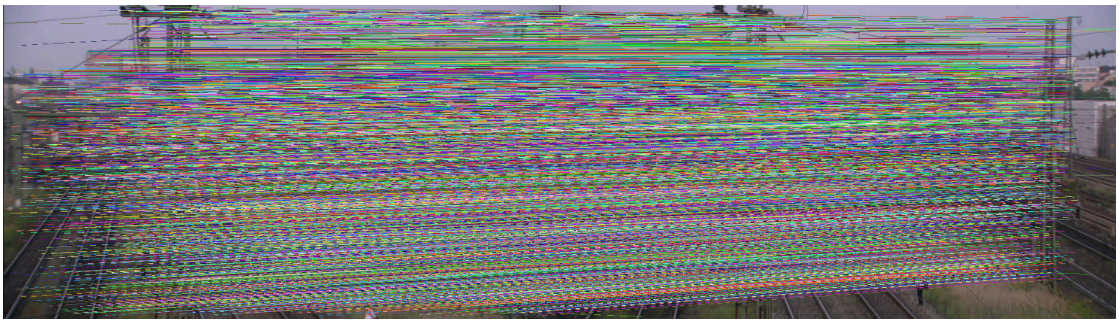
Experimental results for the coding experiments are reported in Table 5.7, the first set (1) of sequences corresponds to targeted sequences with known affine content, other sequences are placed in the second set (2). For comparison, we also introduce a global motion estimator as

(a) Reference frame.

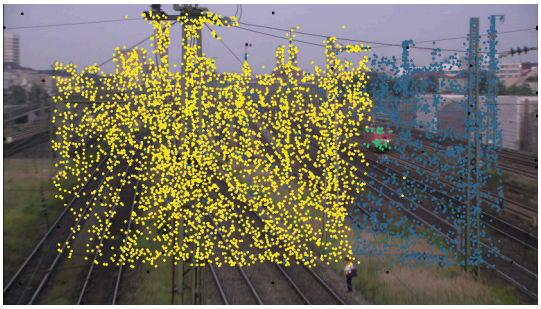
(b) Current frame.



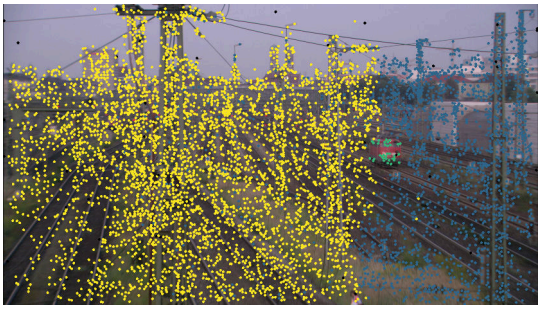
(c) Super-pixel segmentation of the current frame.



(d) Keypoint matching.

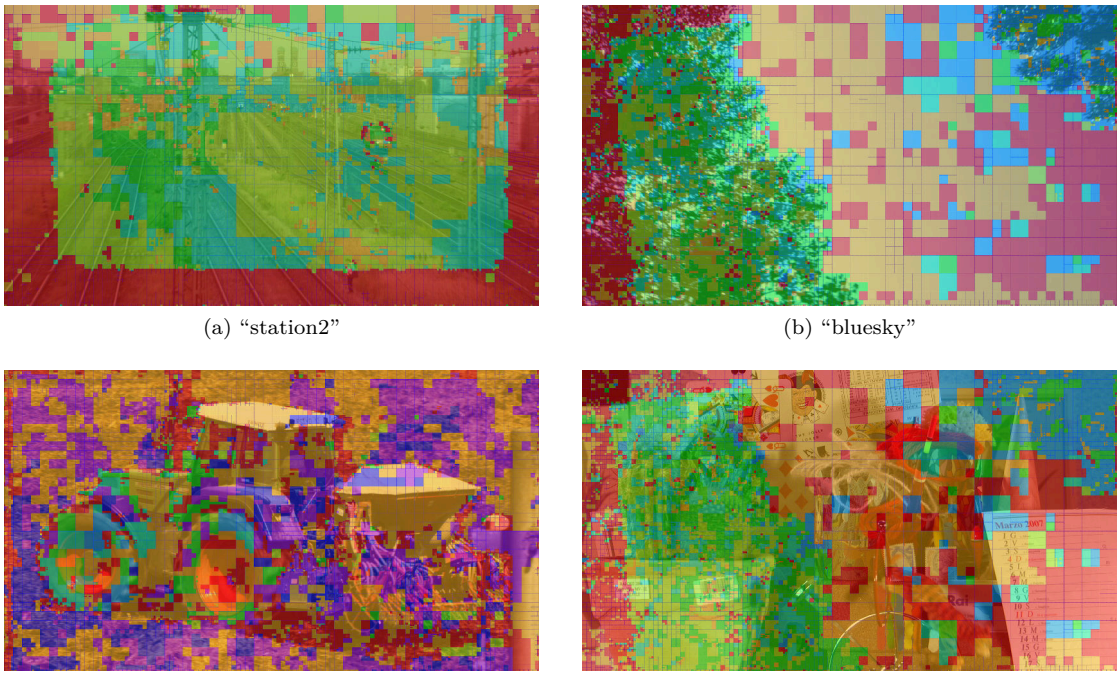


(e) Matched region keypoints (reference frame).



(f) Matched region keypoints (current frame).

Figure 5.11: Overview of the region-based inter-prediction scheme on the test sequence "Station2", which features a large camera zoom. The "footprint" of the zoom motion is clearly visible when comparing the location of the matched keypoints of the reference and current frame.



(a) “station2” (b) “bluesky”

(c) “tractor” (d) “Cactus”

Figure 5.12: Region-based prediction mode usage. Red and orange blocks are coded with classical intra- and inter-prediction, respectively. Predictions made by our mode are depicted with the other colors.

a second baseline, estimated with a classical SIFT+RANSAC approach. BD-rates reductions are presented for the global motion estimator (GLB), the global motion estimator with a global luminance compensation (GLB-L), the proposed region-based approach (RB) and the region-based approach with the luminance compensation (RB-L).

First, one can note that the GLB scheme brings an improvement of -2.88% over the default translational motion models of HEVC on the targeted sequences, highlighting the need for more complex motion prediction models. On the whole dataset, improvements go up to -11.75% with a mean BD-rate reduction of -2.16%.

The proposed region-based prediction mode achieves a greater improvement, with an average gain of -3.04%, up to -12.13%. Most of the sequences benefit from the multiple models prediction, with an average improvement of -0.88% over the single model mode. Although the gains are limited for most sequences, videos with affine motion display significant gains such as “station2” (-12.13%), “LakeWalking” (-9.26%) and “bluesky” (-4.32%).

The efficiency of the luminance compensation is low in the context of the global estimator, with only a marginal improvement of -0.11%, whereas a higher gain of -0.33% can be obtained with the region-based approach. As the luminance compensation is estimated from the matched regions content and not on the global frame, the estimation is more precise and robust.

Overall, our scheme achieves an average BD-rate gain of -3.37%, with -4.19% on targeted affine sequences and especially -1.52% on the second set.

To illustrate the use of the proposed models by the encoder, we adapted an HEVC bit-stream

Table 5.8: Mean time-complexity increases comparison against the baseline HEVC HM software. Mean values are computed on all the sequences of our test dataset.

Method	**Complexity**	
	Encoding	Decoding
GLB	189.92%	174.10%
GLB-L	190.27%	173.18%
RB	299.93%	196.01%
RB-L	301.45%	198.39%

analyzer to display the use of the mode for each block. Examples are shown Figure 5.12 on 4 sequences. One may note that our prediction tool is enabled for a large number of the blocks within the reference frame "footprint". For example, in the "station2" and "bluesky" sequences, the borders are not available for the prediction as a zoom and a rotation were respectively performed by the camera.

5.5.3 Complexity study

We present here a brief complexity study of the proposed scheme. The prediction tool was implemented in the HM software (version 16.16) without particular optimization.

As it is often the case, the main complexity overhead of the prediction scheme resides on the encoder side. The mean complexity increases are reported in Table 5.8. The mean runtime increase of the RB-L scheme is of ∼300% compared to the default HM encoder. Most of the overhead is spent estimating the region-based models and in the increased RDO loop. Numerous improvements are possible to optimize the region-based estimator, especially the keypoints detection, extraction and matching process.

The complexity on the decoder side has a mean increase of ∼200% for the RB-L method. Again, our implementation is not optimized. For example, we warp a whole frame for each model instead of warping only the selected blocks. Moreover, compared to the encoder, the decoder performs only a few extra operations. The model parameters are first decoded, then the blocks are generated by warping and interpolating the reference frame, and the luminance pixel values are finally corrected. All these operations can be relatively easily optimized for fast processing. However, these optimisations would have required numerous modifications of our implementation in the HM software at the detriment of exploring more research areas. Computing new block coordinates from a homography model has a negligible overhead compared to computing from a classical translational model. We measured an increase run time of 60 nanoseconds for 1920×1080 frames. As the computation is still linear on the input ($O(n)$) in complexity and memory, an hardware implementation would have almost no overhead.

5.6 Conclusion and perspectives

In this chapter, we presented a novel prediction scheme for image sets compression. Unlike other approaches from the state-of-the-art, our scheme features a semi-local geometric and photometric prediction method able to compensate in a region-wise manner distortions between two images. The proposed scheme can significantly improve the rate-distortion performances compared to classical image and video coding solutions, with an average gain of -19.62% compared to HEVC. The approach is also competitive compared to state-of-the-art methods. The overhead complexity of our solution is limited and could be significantly reduced by leveraging efficient implementation of the algorithms involved. Although results were only presented based on the HEVC video codec, the proposed prediction method is agnostic to the video codec, allowing to use existing video coding infrastructures without introducing major modifications. Compare to the method presented in the previous chapter, this region-based prediction is faster because it relies on a semi-local prediction, and is more robust as there is no enforced global compensation model that could distort the reference in regions not affected by this global motion.

Interesting challenges still remain, such as exploiting multiple reference frames and ascertaining the scalability of cloud-based image compression techniques. Furthermore, current cloud-based image compression solutions rely on classical content-based image retrieval systems, designed for semantic retrieval. Adapting one of these schemes for compression applications would provide better reference frames for the prediction and ultimately improve the bit-rate distortion performances.

The region-based inter-prediction method was then adapted for video compression. The efficiency of the proposed solution was demonstrated against state-of-the-art video coding tools on multiple sequences with complex motion types, with an average gain of -3.37% over HEVC. The region-based model estimation efficiency over a single global motion model was also demonstrated. The complexity of the prediction mode is limited, especially on the decoder side, with respect to the gains that can be obtained. Although the region-based prediction is currently limited to one reference frame, it could be further extended to use more frames from the reference pictures buffer. Improving the speed and the robustness of the prediction also constitutes an important future work.

Several future works can be considered to further improve our prediction scheme in this context. The SIFT detector and keypoint extractor could be replaced with a faster detector/keypoint extractor. In the presented video compression context, the accuracy of the SIFT algorithm is excessive with respect to the kind of distortions displayed between consecutive frames. Algorithm such as FAST [49] or BRIEF [108] could be considered as replacements. The keypoints extraction and matching could be performed on a lower scale for faster processing, as most of the keypoint extraction algorithms are robust to scale changes to some degree. For now, the algorithm estimates homography models. Affine or similarity transforms might be sufficient in some cases. By reducing iteratively the degrees of freedom until a stable estimation is obtained, the side information cost would be reduced and the robustness and thus quality of the prediction would be improved.

Chapter 6

Learning-based models for inter-prediction

6.1 Introduction

This chapter focuses on deep learning based architectures for video compression. Originally inspired by the human brain structure, neural network architectures have been successfully proposed in the early nineties for solving computer vision tasks such as digits classification [109]. However, the proposed architectures remained quite small in the number of coefficients and layers, constrained by the high computing power required to train such networks. With the increasing availability of computing power, and especially the rise of powerful GPU devices, a deep architecture was first successfully proposed in 2012 by Alex Krizhevsky [110]. The aforementioned "Alexnet" architecture far exceeded the state-of-the-art results on classification benchmarks.

Inspired by the impressive results obtained for image classification tasks, many researchers then proposed deep neural networks architectures for other applications. So far, numerous applications have benefited from deep learning approaches. Tasks such as image classification, image segmentation, style transfer, have obtained the most striking results. Deep learning architectures have also been studied in other fields with great success, such as text processing, audio processing, reinforcement learning, etc...

Recently, deep neural networks have been proposed to solve video interpolation and extrapolation problems. Given a certain number of frames, a network can be trained to predict the next frame(s), or missing intermediate frames. Such networks have been successfully trained for frame extrapolation and/or interpolation. Several approaches have been studied: fully convolutional neural networks [68, 111], transform-based networks [69], adaptive convolution based network [65, 66], optical flow based [67, 74, 77]. These network architectures are promising for video compression applications as they may be used as an additional prediction tool for *hierarchical* and *low-delay* coding, where interpolation and extrapolation network would be respectively useful.

This chapter presents a study of current state-of-the-art deep neural networks for frame interpolation in a video compression context. Deep neural networks were integrated alongside the HEVC video codec for frame inter-prediction. Bit-rate distortions results were collected on classical test sequences. A qualitative study of the interpolation results is also presented.

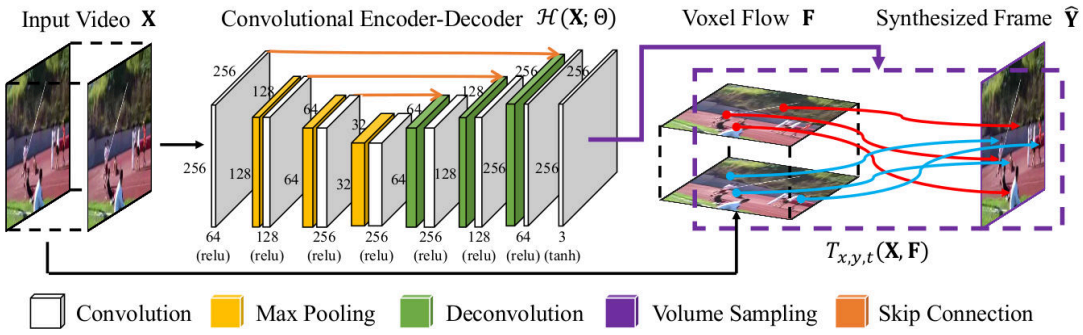


Figure 6.1: *Deep Voxel Flow* architecture [67].

6.2 Deep neural networks for frame interpolation

This section presents the studied deep learning architectures proposed for frame interpolation. These architectures were selected as they have shown to have state-of-the-art performances and their implementation and trained weights were provided at the time of writing this thesis.

6.2.1 Deep Voxel Flow - *DVF*

Architecture The Deep Voxel Flow network (*DVF*) [67] was the first deep neural architecture proposed for frame interpolation which relied on an unsupervised optical flow learning. This network is trained by optimizing directly the reconstruction quality of the interpolated frames. The architecture is a form of convolutional auto encoder with skip connections, to maintain spatial coherence (see Figure 6.1). The skip connections are used here as up-sampling and concatenation operations. The last layer of the network is a non-trainable layer applying a trilinear interpolation function. This constrains the network to generate a *voxel flow*, *i.e.* a per pixel optical flow and a masking temporal weight. The network will thus learn a form of optical flow without supervision. The mask terms is used to handle occlusions and missing objects. For input frames of dimensions $H \times W$, the *voxel flow* will have a size of $H \times W \times 3$, *i.e.* a matrix of (dx, dy, dt). By making the assumption that the motion is temporally linear between the two input frames I_0 and I_1, their respective flow can by computed as:

$$F_0 = (x - dx, y - dy), \text{ and } F_1 = (x + dx, y + dy) \tag{6.1}$$

The interpolated intermediate frame can then be obtained as:

$$I_{0.5} = W \circ \mathcal{B}(I_0, F_0) + (1 - W) \circ \mathcal{B}(I_1, F_1) \tag{6.2}$$

with $\mathcal{B}$ the bilinear interpolation function and W the temporal mask composed of all dt where $W_{i,j} = [0, 1]\ \forall i, j$. The last convolutional layer uses the tanh function as activation. The (dx, dy) values are then denormalized according to the image dimensions, and the (dt) values are scaled to be in the range $[0, 1]$.

Loss functions The network is trained with the l_1-norm, which minimizes the per-pixel difference, and a spatial and temporal coherence regularization term. Total variation regularisation terms are computed separately on the motion term components of the *voxel flow* (dx, dy), and the temporal weight (dt). As explained in the introduction of this manuscript, the l_1-norm is

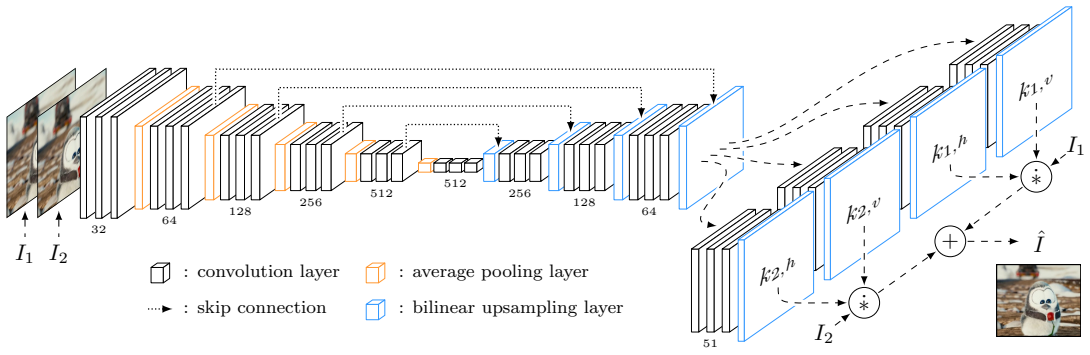


Figure 6.2: *SepConv* architecture [65].

often preferred over the l_2-norm as it leads to less blurry results [68]. The regularization terms are used to enforce spatial and temporal coherence of the *voxel flow*. The total variation loss of a matrix M is derived as:

$$l_1 = \|\hat{I} - I\|_1 \tag{6.3}$$

$$L_{tv} = \sum_{i,j} \| |M_{i,j} - M_{i-1,j}| - |M_{i,j} - M_{i,j-1}| \|_1 \tag{6.4}$$

The objective function used to train the DVF network is a weighted sum of the reconstruction loss and coherence regularization terms:

$$\mathcal{L} = l_1 + \lambda_1 \cdot L_{tv}(dx, dy) + \lambda_2 \cdot L_{tv}(dt) \tag{6.5}$$

With λ_1 and λ_2 empirically set to 0.01 and 0.005 [67].

6.2.2 Adaptive Separable Convolution - *SepConv*

Architecture The adaptive separable convolution approach (*SepConv*) was proposed by Niklaus *et al.* in [65]. Building on their previous work on adaptive convolution [66], they introduced a frame interpolation network relying on separable convolutions. A classical approach of frame interpolation network is to first learn a form of optical flow, describing the per-pixel displacement, and then warp and interpolate the input reference frames to generate an intermediate frame. Niklaus *et al.* proposed instead to train a network to predict local one-dimensional convolution parameters. The frame interpolation is performed as a local convolution over the input frames.

The network architecture is presented in Figure 6.2. The network is based on a classical Convolutional Auto-Encoder (CAE) part, with skip connections to maintain a spatial coherence. The skip connections do not use concatenations as in the *DVF* approach but rather the previous output is up-sampled, convolved and added to the skipped values. This type of architecture was originally proposed for segmentation and is known as U-Net [80]. The main network extracts a dense features map ($H \times W \times N$) which is then forwarded to four sub-networks that estimates the four one-dimensional kernel of 51 pixels ($H \times W \times 51$). These sub-networks estimate the four horizontal and vertical one-dimension kernels that will be used to convolve the two input images. The vertical and horizontal kernel pairs are then used to convolve their respective input frames. Finally the convolutions outputs are summed to obtain the final interpolated frame. By learning directly the convolution filters in a dense-wise manner, the network is able to learn end-to-end the interpolation and synthesis operations.

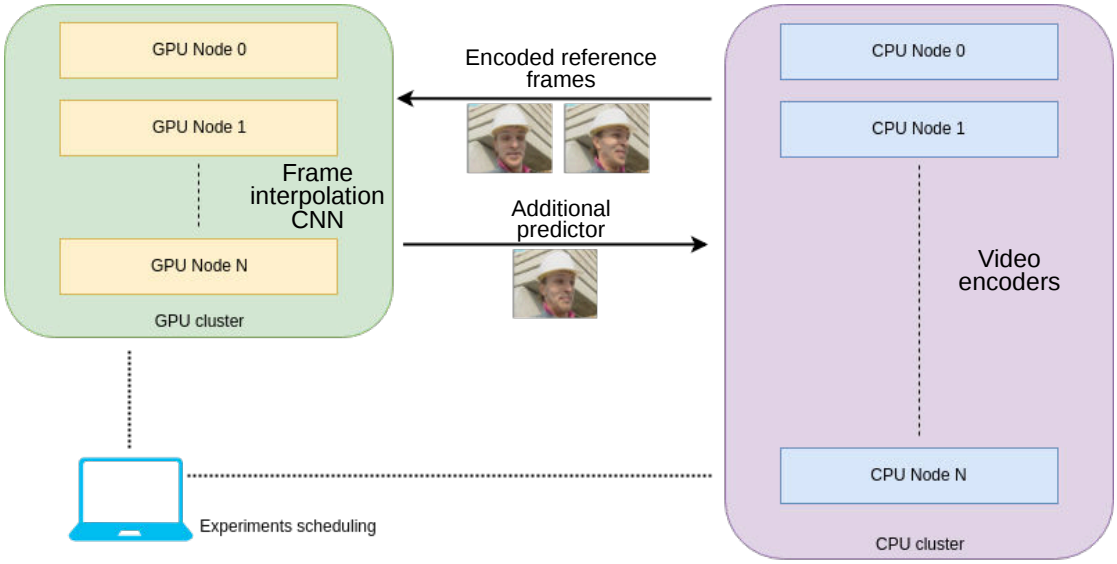


Figure 6.3: Illustration of the experimental setup for running the compression experiments.

Loss functions Two versions of the networks were trained with a loss function measuring the difference between the original ground truth image I and its prediction $\hat{I}$. The first version is trained with the l_1-norm. The second version of the network is trained with a perceptual loss, which tries to minimize the visual differences. The features reconstruction loss l_f is based on the VGG-19 network [75] and was shown to constitute a good perceptual similarity metric [112]. The features are extracted on the `relu4_4` layer. With ϕ the features extraction function, the l_f-norm is then defined as:

$$l_f = \|\phi(\hat{I}) - \phi(I)\|_2^2 \tag{6.6}$$

6.3 Application to video compression

6.3.1 Experimental setup

The experimental setup is represented in Figure 6.3. To benefit from the faster processing speed of GPU devices, the neural networks were run on a GPU cluster and the video encoders on a CPU cluster. The network input frames and the interpolated frames are exchanged over the network. Compared to the computing times required by the encoder, the transmission time over the network is negligible.

For a real world application, the network would have to be implemented in hardware alongside the classical encoder and decoder implementation, which remains an unsolved limitation of deep neural networks for embedded applications.

6.3.2 Coding scheme

To test the networks in a video compression context, the HEVC HM reference software [46] was modified. For each frame hierarchically coded in the GOP, an extra reference frame, interpolated by the network, is appended to the default reference frames list and can be selected as reference

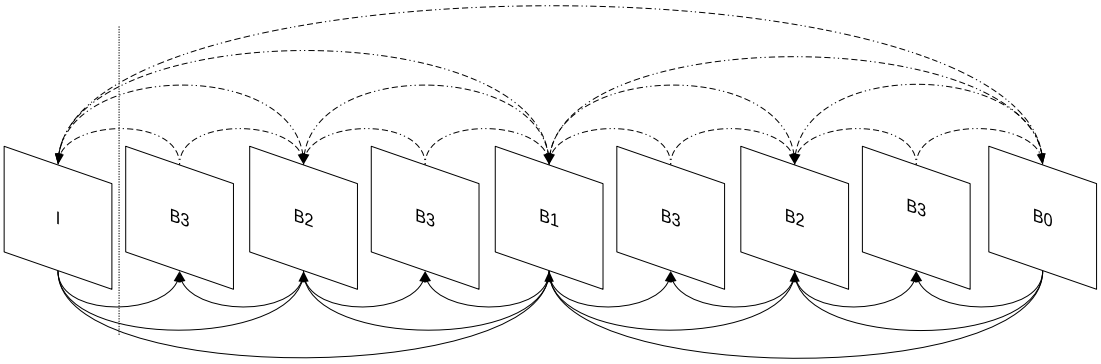


Figure 6.4: Coding scheme: illustration for a GOP of 8 frames with the default *random access* configuration. The dashed lines on top represent the default references, the straight lines below represent the extra reference from the interpolation.

by the rate-distortion optimization process. This new GOP structure is represented in Figure 6.4. For example the *B1* frame can be predicted with the CNN interpolated reference from the *I* and *B0* frames.

Using extra reference frames has two advantages over fully implementing a new inter-prediction mode in HEVC. The implementation is relatively less difficult, but mostly it allows the encoder to use additional inter-prediction modes over the interpolated reference. Although the best configuration would be to only have blocks with SKIP mode as prediction mode (only motion vector predictions, no residual), MERGE or even INTER over the interpolated frames can still be beneficial compared to a prediction from the default reference frames.

To support the use of extra reference frames, several flags needs to be inserted into the bitstream. The decoder will then re-interpolate the selected frames when needed, and thus needs to know if an extra reference frame was used as prediction or not for each coded block.

At the frame level, a flag is first set to signal the use or not of the extra reference frames. At the Prediction Unit (PU) level, an integer is added to the motion vector information to store the use of the extra reference frame used. The sentinel value 0 is set as a signaling method to inform the decoder that no extra reference frames are used for the current PU. The interpolated frame flag is entropy coded with CABAC in order to reduce the syntax size. A CABAC context is also determined from the top and left PU blocks, if they are available for the current block. It should be noted that no specific study was performed on the CABAC context or its initialization, study that would be beneficial once more statistics about the proposed method usages are obtained.

6.3.3 Rate-distortion results

The coding experiments are performed on the common test sequences [46]. The HEVC HM software version 16.16 was used for all the experiments. The rate-distortion performance are computed with the Bjontegaard metric [18] using the common 22, 27, 32, 37 Quantization Parameter (QP). Unless specified otherwise, the PSNR is reported for the Y channel only. The default HM *random access* configuration mode [46] is used as a baseline for the following tests. The default fixed GOP size of 16 is used (a GOP of size 8 is used in Figure 6.4 for clarity). Experiments are run on sequences of the classes B, C and D of the CTC [46], affine sequences class [44] and the legacy *Foreman* sequence. HEVC is used as the baseline reference.

To provide a thorough comparison of multiple frame interpolation methods, results for a

Table 6.1: Summary of the compared frame interpolation methods, classical and deep-learning based methods are studied. Some methods use an intermediate unsupervised optical flow estimation for the interpolation. The interpolation function may then be a classical bilinear filter or a learned interpolation function.

	Method	**Flow estimation**	**Interpolation type**
Classical	CeLiu Flow	Yes	Bilinear
Deep	*SepConv-l1*	No	Custom
	SepConv-lf	No	Custom
	Deep Voxel Flow	Yes	Bilinear

classical optical-flow method [113] are also presented. A summary of the different characteristics of the compared methods is presented in Table 6.1. The default parameters were used for the classical optical flow based method (see *CeliuFow*). *SepConv-l1* is trained with the l_1-loss, and is trained with a features-based loss. *SepConv-lf* is trained with a features-based loss.

The BD-rate performances are reported in Table 6.2. First, one can note that all the experiments bring improvements over the HEVC codec. The best performances are obtained for the *SepConv-l1* method with a mean BD-rate reduction of -2.46%, compared to -2.38% for the *SepConv-lf* network and -0.89% for the *CeliuFow* method. The superior performances of the *SepConv* architecture can be explained by the rigorous training performed on a carefully selected database of sequences. Moreover, the predicted kernels have a size of 51 pixels, and as such can handle relatively larger motions.

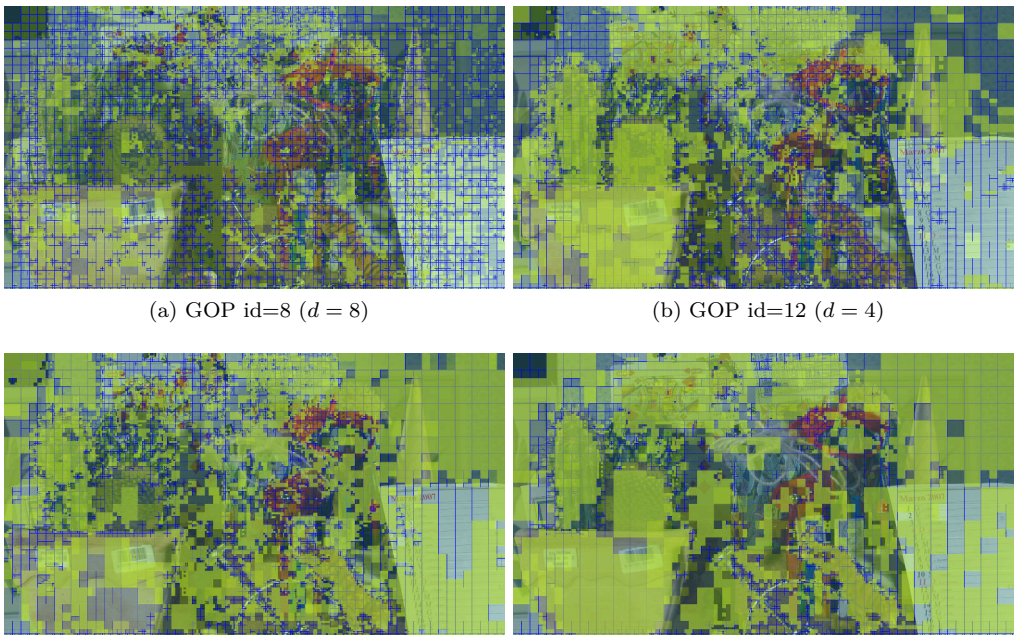
The *SepConv-l1* method outperforms the other methods. This network was specifically trained with the $l1$-norm, which minimized the interpolation error energy, explaining the greater average performance of -0.08% over the *SepConv-lf* network. Surprisingly, the *SepConv-lf* performs better on three of the sequences. Both *SepConv* approaches outperform the classical optical flow approach by -1.45%. However the *Deep Voxel Flow* network has the worst bit-rate distortion performances, with a mean gain of -0.09%. Some values could also not be reported due to memory limitations in the published implementation. It is also important to note that the deep architectures were trained on RGB images, whereas HEVC encodes YCbCr frames with a spatial 4:2:0 sampling. The input frames need to be up-sampled (chroma wise) and be converted to RGB before entering the network, the inverse operation is performed before the encoder. There is some loss of information during these conversions. Training the networks to process YCbCr frames with a 4:2:0 sampling would lead to better results.

The best bit-rate reduction, -7.36%, is obtained on the *"Cactus"*, which can be explained by the large number of small object motions in this sequence. Examples of the encoder selection of the proposed interpolated reference by the network are shown in Figure 6.5. Estimating and compensating efficiently these small local motions require a lot of syntax signaling (quad-tree splitting, motion vectors information, residual). This costly prediction is avoided by interpolating directly the intermediate blocks with the approach.

Affine sequences benefit less from the interpolation methods as they mostly display large global scene motions (due to camera zooms or shakes for example), which are more difficult to estimate for these methods, trained for small local motions estimation. The region-based approach presented in the previous chapter is more performant on these affine sequences with an average bit-rate reduction of -3.31% compared to -0.84% here. It would be interesting to investigate recently proposed deep architectures designed for homography estimation [114, 115].

Table 6.2: BD-rate performances comparison for different frame interpolation methods compared to HEVC.

Class	**Sequence**	**SepConv-l1**	**SepConv-lf**	**DVF**	**CeLiu Flow**
B	BQTerrace	**-6.38%**	-5.94%	–	-2.77%
	BasketballDrive	**-1.43%**	-1.05%	–	-0.35%
	Cactus	**-7.36%**	-7.07%	–	-4.56%
	Kimono1	**-2.05%**	-1.71%	–	-0.23%
	ParkScene	**-2.36%**	-2.14%	–	-0.41%
C	BQMall	**-3.44%**	-2.89%	-0.10%	-0.34%
	BasketballDrill	**-3.87%**	-3.77%	-0.71%	-1.17%
	PartyScene	-2.53%	**-3.08%**	-0.14%	-0.37%
	RaceHorses	**-1.30%**	-1.01%	-0.03%	-0.47%
D	BQSquare	-3.31%	**-5.48%**	0.10%	0.06%
	BasketballPass	**-2.93%**	-2.51%	-0.13%	-0.34%
	BlowingBubbles	-2.08%	**-2.29%**	-0.13%	-0.24%
	RaceHorses	**-2.21%**	-1.84%	-0.06%	-0.29%
Affine	CStoreGoods	**-0.45%**	-0.29%	–	-0.25%
	DrivingRecorder1	**-1.37%**	-0.93%	0.04%	-0.07%
	DrivingRecorder2	**-1.24%**	-1.03%	0.01%	-0.24%
	LakeWalking	**-0.08%**	0.00%	0.03%	-0.05%
	ParkSunny	**-1.34%**	-1.34%	–	-0.82%
	ParkWalking	**-0.54%**	-0.32%	–	-0.03%
Other	foreman	**-2.92%**	-2.82%	0.08%	-0.32%
Mean		**-2.46%**	-2.38%	-0.09%	-0.89%



(a) GOP id=8 ($d = 8$) (b) GOP id=12 ($d = 4$)

(c) GOP id=14 ($d = 2$) (d) GOP id=15 ($d = 1$)

Figure 6.5: Examples of modes selection on the *Cactus* sequence at QP=22. Blocks in green are encoded from the interpolated reference by the network. Frames with a smaller frame distance (d) between the references benefit more from the additional reference, as the motion range is smaller.

Table 6.3: Comparison of the mean interpolation times for each method, on the three classes of the CTC sequences [46].

Method	**Platform**	**Interpolation time (seconds)**		
		Class B	Class C	Class D
CeliuFow	CPU	52.53	9.90	2.43
DVF	GPU	–	–	0.13
SepConv	GPU	0.99	0.30	0.10

Table 6.4: Comparison of the number of trained coefficients/weights for each network.

Architecture	**Number of coefficients**
DVF	3.8 M
SepConv	21.68 M

6.3.4 Qualitative results

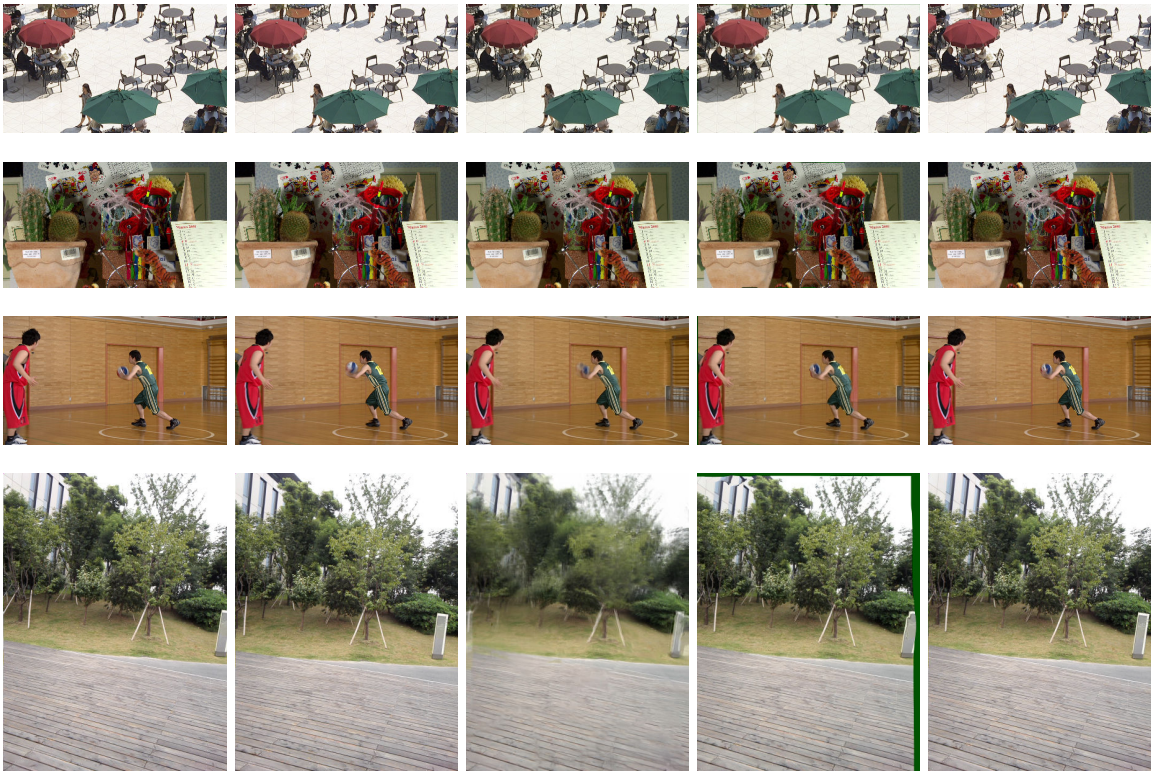
Qualitative results are presented in Figure 6.5. Interpolated frames are computed for different distances in a group of pictures. The compression experiments use the standard GOP size of 16. As such, this section provides visual results for interpolations obtained for the four possible GOP distances between reference frames: 1, 2, 4, and 8. Results are shown for four sequences of the test set: *"BQSquare"*, *"Cactus"*, *"BasketBallPass"*, and *"LakeWalking"*.

One might note that small local motions are well interpolated, for example the people walking in *"BQSquare"*, the objects rotating in *"Cactus"* and the players and their ball in *"BasketBallPass"*. The interpolation methods perform better for small GOP distances (1, 2), indeed when the GOP distance between reference frames is too large, blurry outputs are generated. This might be explained by the fact that distant reference frames do not have a field motion that is still locally linear, as assumed by the networks, for example the curved trajectory of the basket ball in the *"BasketBallPass"* sequence. Moreover both methods struggle with the *"LakeWalking"* sequence which displays strong scene motion due to the camera movements (the lack of padding for the *CeliuFow* method is also clearly visible), and also illumination changes.

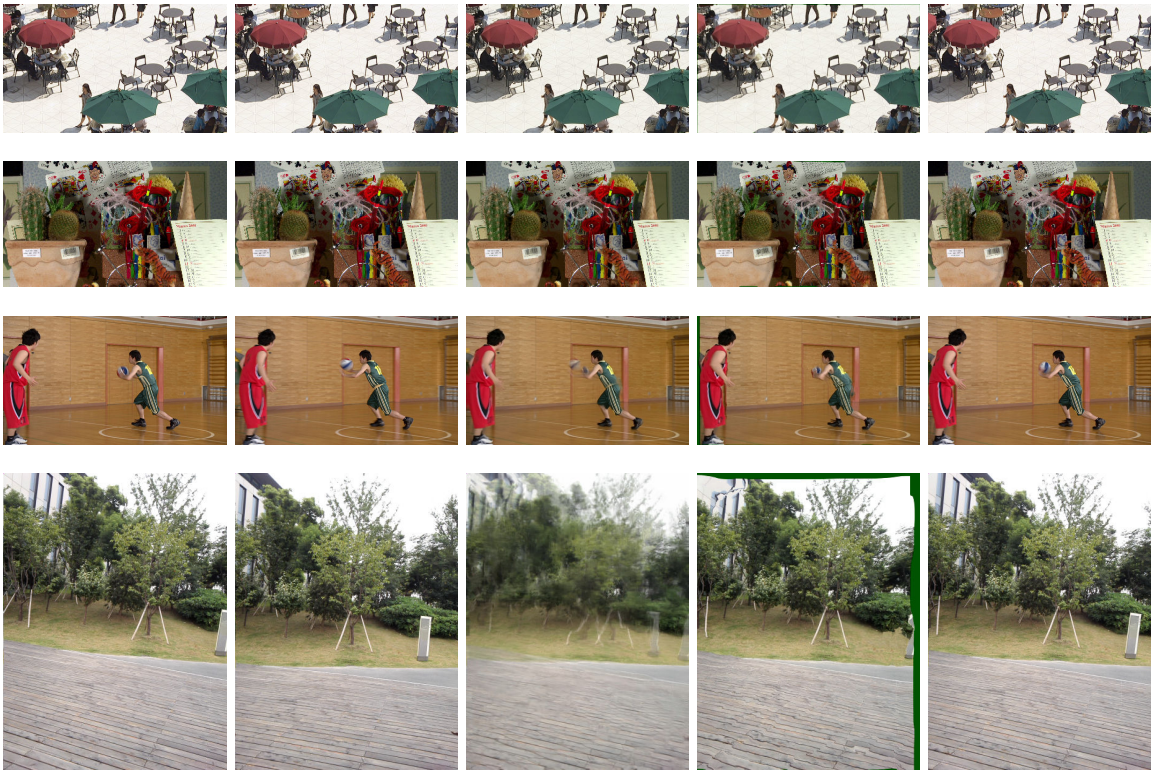
6.3.5 Complexity study

When using deep convolutional networks in conjunction with classical architectures, it is difficult to provide a meaningful complexity study as some algorithms are designed to run efficiently on GPU and not on CPU. Instead two studies are reported in this section. The interpolation times for each methods are compared for 3 classes from the CTC test sequences. The experiments were run with a single core on an Intel Xeon X5650 CPU, and a Nvidia GeForce GTX 1070 GPU. The number of trained coefficients (the weights) for the deep neural network architectures are also reported in Table 6.4.

The *SepConv* approach is at least 25 times faster than the *CeliuFow* method, and about 1.3 times faster that *DVF*. It benefits from being run on a GPU, and an efficient CUDA implementation. However this network requires 21 million parameters, which takes an approximate size of 82 mega-bytes on the disk, and a single forward pass of a 256x256 pixels patch requires around 216 mega-bytes of memory during the execution.



(d) GOP distance: 1



(h) GOP distance: 2

(l) GOP distance: 4

(p) GOP distance: 8

Figure 6.5: Qualitative evaluation of the different interpolation methods for different GOP distance. From left to right: first input frame, second input frame, interpolated frame via *SepConv*, interpolated frames via *CeliuFow*, groundtruth frame. Sequences (top to bottom): *"BQSquare"*, *"Cactus"*, *"BasketBallPass"*, and *"LakeWalking"*.

6.4 Conclusion and perspectives

This last chapter presented experiments on using deep neural networks for inter-prediction. The efficiency of deep architectures was demonstrated against classical methods and the latest HEVC video codec. This work constitutes a first step in leveraging deep networks for frame inter-prediction. Unfortunately the tested networks could not be fine-tuned on our video compression datasets as only the forward pass implementations were published. Compared to the approaches presented in the previous chapters, the current method is even more generic as it is not limited to parametric motion models. The robustness is also improved as it relies on a deep-learning approach which allow to learn a good generalization of the motion estimation.

A lot of research remains to be performed to improve the design of deep neural networks for video compression. Networks are usually trained, and operates, on 32-bit floating point values. However video codec standards rely on integer operations for the sake of reproducibility. Works have already been proposed for implementing network coefficients quantization [116], such schemes will need to be researched and adapted for video compression. The interpolation filters used by the network should also be tailored for video inter-prediction.

A closer integration in the codec would bring improvements thanks to the RDO loop. Moreover the predicted motion fields could be re-used by the encoder. The estimated motion range for these deep architectures is limited to small local motions for now, which limits the efficiency on affine sequences. Finally, a large and pertinent database is crucial for training deep learning based architectures, such a database built for video compression applications would be useful for the compression community.

Chapter 7

Conclusion

The ever-growing production and consumption of image and video media requires constant innovation in the image and video compression domain. Thanks to new communication channels (optical fiber, 4G, 5G) more bandwidth is becoming available to end-users. However this bandwidth increases can not cope with new media usages and formats. Television usage is shifting from terrestrial broadcast to internet or cable unicast streaming. An increasing amount of user generated contents is being uploaded and consumed on social network platforms. Novel formats and usages such as 360° videos, Augmented Reality, Virtual Reality, require an even more significant amount of data.

This thesis explored novel inter-prediction tools for image sets and video compression to cope with this growing demand. Inter-prediction constitutes an essential part of the traditional video compression framework. By capturing redundancies, inter-prediction methods are able to significantly reduce the bit-stream size required to encode videos. Inter-prediction techniques have also been leveraged for compressing photo-albums or image databases where there is a high similarity between images, which can be compressed as consecutive video frames. This work introduced and studied several inter-prediction approaches modeling the distortions between frames, with methods of increasing generalization capacity.

The first contribution chapter introduced a novel inter-prediction method for image sets compression, based on a global geometric and photometric compensation followed by local, block-based, compensations. Video codecs are traditionally designed to compensate the motions between frames by estimating translational motion vectors. However such models can not compensate correctly a number of real world distortions that can be seen between similar images in a database. Such distortions usually include large geometric changes (*e.g.* zooms, rotations, change of viewpoint or focal length) and photometric changes (*e.g.* difference of illumination, weather conditions, color balance). The proposed global photometric and geometric compensation scheme is able to compensate for the large distortions, then a compensation is performed locally to refine the prediction at the block level. Experimental results demonstrate that the proposed scheme can significantly improve the compression performance compared to approaches based on video-coding. However, the proposed method is computationally expensive.

In the second chapter, a region-based inter-prediction scheme is proposed. By estimating geometric and photometric compensation models at a semi-local level, the method is able to efficiently leverage redundancies between frames. Input frames are first segmented with superpixels, which allow to cluster similar pixels in a local search window. Homography transform models are then recursively estimated and refined via a graph-cut approach. The most prominent models are extracted and retained. A region-based segmentation is obtained by assigning a

geometric transform to each super-pixel. Photometric compensation models are finally estimated for each region. Two models are proposed: a simple luminance scale offset correction, and a more complex piece-wise spline correction that can handle stronger color disparities. The method was tested and validated on a large dataset of images. Significant improvements are obtained compared to traditional video coding tools, with a mean bit-rate reduction of -19.62%. The proposed solution is also fast and competitive against state-of-the-art methods.

The proposed region-based inter-prediction method is then adapted for a video compression context. Indeed, the traditional translational motion models of video codecs are often not sufficient to handle real world complex motions such as zooms, rotations, pans. The region-based method is adapted and implemented alongside the HEVC video codec. Bit-rate distortion improvements are obtained compared to classical solutions, especially for sequences displaying affine motions with a mean average gain of -3.37%. The semi-local geometric and photometric prediction is also validated against a simpler global prediction scheme (-3.37% compared to -2.27%).

In the last chapter, deep learning based methods for inter-prediction are studied. Deep neural networks have shown striking results for a large number of computer vision tasks in recent years. Deep neural networks have also been designed for image prediction, and demonstrated results close to state-of-the-art classical image codec. Fewer works studied deep architectures for video compression. However, networks were trained successfully for frame interpolation tasks, which can be seen as a form of inter-prediction. In this chapter, such networks are adapted and integrated into a classical video codec. Experimental results are gathered for the traditional bi-predictive hierarchical coding structure of video codecs. Bit-rate reductions over classical inter-prediction techniques in the HEVC codec are demonstrated (-2.29% on average). These first results highlight the potential of deep architectures for video compression applications, especially thanks to their good generalization ability.

Future works and perspectives

Several approaches can be considered to extend the works presented in this thesis.

Parametric motion models for video compression The global and region-based inter-prediction schemes would require further research to be included in a video compression standard. The keypoints detection and extraction is a key step of these methods. The SURF and SIFT algorithms were used as they were both proven for their robustness and accuracy. However, faster algorithms could be used in a video compression context. For example the AV1 codec selected the FAST descriptors. Keypoints detections algorithms are also robust to scale (to some extent). As such, processing the proposed inter-prediction schemes on a down-sampled version of the input frames could reduce the processing times without losing too much prediction accuracy. The keypoint matching could benefit from geometric matching constraints for faster processing. In the video compression context, the keypoint matches search could be restricted to a local search window between consecutive frames.

Concerning the robustness of the geometric estimations, only homography models were considered in this thesis. However, models with fewer degrees of freedom, such as affine transforms (6 degrees) or similarity transforms (4 degrees) could be considered. A set of possible transform models could then be iteratively tested until a sufficient prediction is obtained. To extract the geometric transforms, a graph cut is recursively applied. The loss function of this graph cut requires several weighting parameters, which were estimated on a training database. Although good performances were obtained with this parameters set, it is still uncertain how to compute

the best values for these parameters.

More research could also be conducted towards a tighter integration into a video codec to improve the encoding performances. For example, the graph cut approach uses a label cost term which restricts the number of models to be selected. A better approach could be to integrate a loop closer to the RDO process, which would take both the distortion and the bit-rate (residual and syntax) into account. The proposed method relied on bilinear interpolation to warp the input frames with the estimated geometric models. However, specific interpolation functions have already been developed and integrated in video codecs. For instance HEVC has two separate interpolation functions, for luma and chroma. Although HEVC is the latest MPEG standard, ongoing work under the Joint Exploration Model (JEM) already provides a 35% bit-saving over HEVC [117], further experiments with this experimental codec could be performed to validate the proposed methods.

Image retrieval for image compression Image sets compression schemes rely on retrieving suitable reference frames from a canonical database. Content based image retrieval methods have already been studied for several applications such as reverse image search. However these methods are designed primarily to find semantically similar images (*e.g.* a "car", a "tree"). As such the retrieved images may not be the best reference frames. Adapting image retrieval schemes for compression applications thus constitutes an important future work.

Deep learning for video compression

- Current limitations: A lot of research remains to be performed to improve the design and training of deep neural networks for video compression. One of the main challenge is to integrate a loss function during training which emulates the rate-distortion optimization process. In this manuscript, the $l1$-norm was used, which is known to minimize both the distortion and the residual energy. However, a loss function designed to reduce the distortion and the bit-rate would greatly improve the performances. Estimating the exact required bit-rate may not be possible for a network, especially when considering complex non-differentiable arithmetic coder such as CABAC. As such simpler approximations could be performed on the residual. The entropy coding is performed on quantized coefficients in the transform domain, which is a non-differentiable operation. Research has shown that the quantization can be backward propagated as a identity function [118] or a uniform noise between the quantization steps [59]. The interpolation filters used by the network should also be tailored for video inter-prediction.

- Future challenges: Networks are usually trained, and operate, on 32-bit floating point values. However video codec standards rely on integer operations for the sake of reproducibility. Works have already been proposed for implementing network coefficients quantization [116], such schemes will need to be researched and adapted for video compression. There is also some uncertainty about the feasibility of using such architectures on the decoder side. The required energy consumption and the network size (which translates into silicon surface) is not currently practical for real world usage. Also, these architectures can not yet be run efficiently on CPUs due to the high complexity requirements. Numerous challenges still need to be explored to develop efficient and practical deep neural networks for video compression. Such developments should be interesting to follow for both the video compression and computer vision communities.

Author's publications

Articles

Conference papers

J. Bégaint, D. Thoreau, P. Guillotel and M. Türkan, "Locally Weighted Template-Matching Based Prediction for Cloud-Based Image Compression", 2016 Data Compression Conference (DCC), Snowbird, UT, 2016, pp. 417–426.

J. Bégaint, F. Galpin, P. Guillotel, C. Guillemot. "Region-based models for motion compensation in video compression", PCS 2018 - Picture Coding Symposium, Jun 2018, San Francisco, United States.

J. Bégaint, F. Galpin, P. Guillotel, C. Guillemot. "Deep inter-prediction for video compression", *Planned*.

Journal paper

J. Bégaint, D. Thoreau, P. Guillotel and C. Guillemot, "Region-Based Prediction for Image Compression in the Cloud", in IEEE Transactions on Image Processing, vol. 27, no. 4, pp. 1835–1846, April 2018.

Patents

4 patent applications pending.

Glossary

ANN	Artificial Neural Network
AR	Augmented Reality
AVC	Advanced Video Coding
BMC	Block Motion Compensation
BME	Block Motion Estimation
CABAC	Context Adaptive Binary Arithmetic Coder
CBIR	Content Based Image retrieval
CNN	Convolutional Neural Network
CTU	Coding Tree Unit
CU	Coding Unit
DCT	Discrete Cosine Transform
DM	Direct Mode
DNN	Deep Neural Network
DST	Discrete Sine Transform
GOP	Group Of Picture
HDR	High Dynamic Range
HEVC	High Efficiency Video Coding
HFR	High Frame Rate
HVS	Human Vision System
JEM	Joint Exploration Project
JPEG	Joint Photographic Experts Group
JVET	Joint Video Exploration Team
K-NN	K-Nearest Neighbour
LLE	Locally Linear Embedding
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
MST	Minimum Spanning Tree
PSNR	Peak Signal-to-Noise Ratio
PU	Prediction Unit
QP	Quantization Parameter
RANSAC	Random Sample Consensus
RDO	Rate Distortion Optimization
RS	Representative Signal
SAD	Sum of Absolution Difference
SAO	Sample Adaptive Offset
SIFT	Scale Invariant Feature Transform

SI	Side Information
SSIM	Structural Similarity
TM	Template Matching
TU	Transform Unit
UGC	User Generated Content
UHD	Ultra High Definition
VR	Virtual Reality
VVC	Versatile Video Coding
WCG	Wide Color Gamut

List of Figures

List of Tables

Bibliography

[1] Facebook, Ericsson, and Qualcomm, "A focus on efficiency," *A whitepaper from Facebook, Ericsson and Qualcomm*, September 2013.

[2] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg2000 still image compression standard," *IEEE Signal Proc. Mag*, 2001.

[3] J. Bankoski, P. Wilkins, and Y. Xu, "Technical overview of vp8, an open source video codec for the web," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1–6.

[4] F. Bellard. (2015) Bpg. [Online]. Available: https://bellard.org/bpg/

[5] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[6] CISCO, "CISCO visual networking index: Forecast and methodology, 20162021." 2016.

[7] F. Bossen, "Mpeg standardisation roadmap," in *Proc. 123th MPEG Meeting*, Ljubljana, SI, July 2018.

[8] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," Tech. Rep., 2000.

[9] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 620–636, 2003.

[10] F. Wu, *Advances in Visual Data Compression and Communication: Meeting the Requirements of New Applications.* Auerbach Publications, 2014.

[11] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[12] M. Wien, "High efficiency video coding."

[13] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[14] M. Flierl and B. Girod, "Generalized b pictures and the draft h. 264/avc video-compression standard," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 13, no. 7, pp. 587–597, 2003.

[15] M. Budagavi, A. Fuldseth, G. Bjøntegaard, V. Sze, and M. Sadafale, "Core transform design in the high efficiency video coding (hevc) standard," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1029–1041, Dec 2013.

[16] I. J. Group, "libjpeg 6.2," http://ijg.org/.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. Simoncelli, "Quality assessment: from error measurement to structural similarity," *IEEE Trans. Image Process*, vol. 13, no. 4, pp. 600–612, 2004.

[18] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," in *ITU-T SG16/Q6 VCEG document VCEG-M33*, Austin, TX, USA, Apr 2001.

[19] K. Karadimitriou, "Set redundancy, the enhanced compression model, and methods for compressing sets of similar images," Ph.D. dissertation, Department of Computer Science, Louisiana State University, Baton Rouge, La, USA, 1996.

[20] R. Zou, O. C. Au, G. Zhou, W. Dai, W. Hu, and P. Wan, "Personal photo album compression and management." in *ISCAS*, 2013, pp. 1428–1431.

[21] Z. Shi, X. Sun, and F. Wu, "Photo album compression for cloud storage using local features," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 4, no. 1, pp. 17–28, 2014.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] D. Perra and J. Frahm, "Cloud-scale image compression through content deduplication." in *BMVC*, 2014.

[25] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

[26] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of GIST descriptors for web-scale image search," in *Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR,*, 2009.

[27] H. Yue, X. Sun, J. Yang, and F. Wu, "Cloud-based image coding for mobile devices - toward thousands to one compression," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 845–857, 2013.

[28] Z. Yuan, P. Yan, and S. Li, "Super resolution based on scale invariant feature transform," in *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on.* IEEE, 2008, pp. 1550–1554.

[29] M. Amintoosi, M. Fathy, and N. Mozayani, "Regional varying image super-resolution," in *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, vol. 1. IEEE, 2009, pp. 913–917.

[30] C.-C. Hsu and C.-W. Lin, "Image super-resolution via feature-based affine transform," in *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on.* IEEE, 2011, pp. 1–5.

[31] H. Yue, J. Yang, X. Sun, and F. Wu, "Sift-based image super-resolution," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 2896–2899.

[32] H. Yue, X. Sun, J. Yang, and F. Wu, "Landmark image super-resolution by retrieving web images," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4865–4878, 2013.

[33] L. Sun and J. Hays, "Super-resolution from internet-scale scene matching," in *Computational Photography (ICCP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–12.

[34] H. Yue, X. Sun, J. Yang, and F. Wu, "Cid: Combined image denoising in spatial and frequency domains using web images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2933–2940.

[35] ——, "Image denoising by exploring external and internal correlations," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1967–1982, 2015.

[36] Y. Zhang, W. Lin, and J. Cai, "Dense correspondence based prediction for image set compression," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2015, pp. 1240–1244.

[37] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 70:1–70:10, Jul. 2011.

[38] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[39] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE transactions on image processing*, vol. 9, no. 3, pp. 497–501, 2000.

[40] H. Watanabe and K. Jinzenji, "Sprite coding in object-based video coding standard: Mpeg-4," in *Proceedings of Multiconference on Systemics, Cybernetics and Informatics*, vol. 13. Citeseer, 2001, pp. 420–425.

[41] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, and H. Watanabe, "Two-stage motion compensation using adaptive global mc and local affine mc," *IEEE Transactions on Circuits and Systems for video technology*, vol. 7, no. 1, pp. 75–85, 1997.

[42] K. Rijkse, "H. 263: video coding for low-bit-rate communication," *IEEE Communications magazine*, vol. 34, no. 12, pp. 42–45, 1996.

[43] H. Huang, J. W. Woods, Y. Zhao, and H. Bai, "Affine SKIP and DIRECT modes for efficient video coding," in *2012 Visual Communications and Image Processing, VCIP 2012, San Diego, CA, USA, November 27-30, 2012*, 2012, pp. 1–6.

[44] H. Chen, F. Liang, and S. Lin, "Affine SKIP and MERGE modes for video coding," in *17th IEEE International Workshop on Multimedia Signal Processing, MMSP 2015, Xiamen, China, October 19-21, 2015*, 2015, pp. 1–5.

[45] J. V. E. T. J. on Future Video Coding, "JVET JEM software," https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/.

[46] F. Bossen, "Common test conditions and software reference configurations," in *Proc. 12th JVT-VC Meeting, JCTVC-K1100*, Shanghai, CN, October 2012.

[47] S. Parker, Y. Chen, D. Barker, P. de Rivaz, and D. Mukherjee, "Global and locally adaptive warped motion compensation in video compression," in *IEEE International Conference on Image Processing, ICIP 2017*, 2017.

[48] "Alliance for open media," http://aomedia.org.

[49] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[50] H. L. Tan, C. C. Ko, and S. Rahardja, "Fast coding quad-tree decisions using prediction residuals statistics for high efficiency video coding (HEVC)," *TBC*, vol. 62, no. 1, pp. 128–133, 2016.

[51] A. Mercat, F. Arrestier, M. Pelcat, W. Hamidouche, and D. Ménard, "Prediction of quad-tree partitioning for budgeted energy HEVC encoding," in *2017 IEEE International Workshop on Signal Processing Systems, SiPS 2017, Lorient, France, October 3-5, 2017*, 2017, pp. 1–6.

[52] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, 2018.

[53] T. Dumas, A. Roumy, and C. Guillemot, "Context-adaptive neural network based prediction for image compression," *arXiv preprint arXiv:1807.06244*, 2018.

[54] Z. Zhao, S. Wang, S. Wang, X. Zhang, S. Ma, and J. Yang, "CNN-based bi-directional motion compensation for high efficiency video coding," p. 4.

[55] R. Lin, Y. Zhang, H. Wang, X. Wang, and Q. Dai, "Deep convolutional neural network for decompressed video enhancement," in *Data Compression Conference (DCC), 2016*. IEEE, 2016, pp. 617–617.

[56] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in hevc intra coding," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[57] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.

[58] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," pp. 1646–1654. [Online]. Available: http://ieeexplore.ieee.org/document/7780551/

[59] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *Picture Coding Symposium (PCS), 2016*. IEEE, 2016, pp. 1–5.

[60] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[61] O. Rippel and L. Bourdev, "Real-time adaptive image compression," *arXiv preprint arXiv:1705.05823*, 2017.

[62] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks."

[63] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[64] C.-Y. Wu, N. Singhal, and P. Krähenbühl, "Video compression through image interpolation." [Online]. Available: http://arxiv.org/abs/1804.06919

[65] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.

[66] ——, "Video frame interpolation via adaptive convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 670–679.

[67] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 4473–4481.

[68] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[69] J. Van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, "Transformation-based models of video sequences," *arXiv preprint arXiv:1701.08435*, 2017.

[70] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE conference on computer vision and pattern recognition (CVPR)*, vol. 2, 2017, p. 6.

[71] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[72] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[73] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[74] J. van Amersfoort, W. Shi, A. Acosta, F. Massa, J. Totz, Z. Wang, and J. Caballero, "Frame interpolation with multi-scale deep loss functions and generative adversarial networks." [Online]. Available: http://arxiv.org/abs/1711.06045

[75] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[76] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," *arXiv preprint arXiv:1712.00080*, 2017.

[77] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," *arXiv preprint arXiv:1803.10967*, 2018.

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[79] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," *arXiv preprint arXiv:1707.07958*, 2017.

[80] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[81] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *arXiv*, 2017.

[82] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[83] Y. Xiong and K. Pulli, "Color correction for mobile panorama imaging," in *The First International Conference on Internet Multimedia Computing and Service, ICIMCS '09, Kunming, Yunnan, YT, China, November 23-25, 2009*, 2009, pp. 219–226.

[84] T. Pouli and E. Reinhard, "Progressive color transfer for images of arbitrary dynamic range," *Computers & Graphics*, vol. 35, no. 1, pp. 67–80, 2011.

[85] M. Alain, S. Cherigui, C. Guillemot, D. Thoreau, and P. Guillotel, "Locally linear embedding methods for inter image coding," in *IEEE International Conference on Image Processing, ICIP 2013, Melbourne, Australia, September 15-18, 2013*, 2013, pp. 1904–1908.

[86] M. Türkan and C. Guillemot, "Image prediction based on neighbor-embedding methods," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1885–1898, 2012.

[87] S. Cherigui, C. Guillemot, D. Thoreau, P. Guillotel, and P. Pérez, "Map-aided locally linear embedding methods for image prediction," in *19th IEEE International Conference on Image Processing, ICIP 2012, Lake Buena Vista, Orlando, FL, USA, September 30 - October 3, 2012*, 2012, pp. 2909–2912.

[88] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European conference on computer vision*. Springer, 2008, pp. 304–317.

[89] H. Shao, T. Svoboda, and L. V. Gool, "ZuBuD — Zürich buildings database for image based recognition," Computer Vision Laboratory, Swiss Federal Institute of Technology, Tech. Rep. 260, March 2003.

[90] J. Heinly, E. Dunn, and J. Frahm, "Comparative evaluation of binary features," in *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Proceedings, Part II*, 2012, pp. 759–773.

[91] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. INSTICC Press, 2009, pp. 331–340.

[92] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[93] F. Zhang and D. R. Bull, "A parametric framework for video compression using region-based texture models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1378–1392, 2011.

[94] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE transactions on Image processing*, vol. 4, no. 6, pp. 832–838, 1995.

[95] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *International Journal of Computer Vision*, vol. 96, no. 1, pp. 1–27, 2012.

[96] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[97] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition, 2012*, 2012, pp. 2911–2918.

[98] A. Harltey and A. Zisserman, *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006.

[99] E. Vincent and R. Laganiere, "Detecting planar homographies in an image pair," in *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces*, 2001, pp. 182–187.

[100] H. N. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International Journal of Computer Vision*, vol. 97, no. 2, pp. 123–147, 2012.

[101] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Optimizing color consistency in photo collections," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 38:1–38:10, 2013.

[102] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs," in *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Proceedings, Part II*, 2002, pp. 541–555.

[103] R. Storn and K. V. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[104] M. Björkman, N. Bergström, and D. Kragic, "Detecting, segmenting and tracking unknown objects using multi-label MRF inference," *Computer Vision and Image Understanding*, vol. 118, pp. 111–127, 2014.

[105] C. Y. Ren, V. A. Prisacariu, and I. D. Reid, "gSLICr: SLIC superpixels at over 250Hz," *ArXiv e-prints*, Sep. 2015.

[106] "Xiph.org test media," http://media.xiph.org/video/derf/.

[107] X. Ma, H. Zhang, Y. Zhao, M. Sun, M. Sychev, H. Yang, and J. Zhou, "Huawei test sequences of UGC feature for video coding development," in *Proc. 22th JVT-VC Meeting*, Geneva, Switzerland, Oct 2015.

[108] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[109] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[110] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[111] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," *arXiv preprint arXiv:1712.09867*, 2017.

[112] R. Zhang, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric."

[113] C. Liu *et al.*, "Beyond pixels: exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[114] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *arXiv preprint arXiv:1606.03798*, 2016.

[115] F. Erlik Nowruzi, R. Laganiere, and N. Japkowicz, "Homography estimation from image pairs with hierarchical convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 913–920.

[116] S. Wu, G. Li, F. Chen, and L. Shi, "Training and inference with integers in deep neural networks," *arXiv preprint arXiv:1802.04680*, 2018.

[117] N. Sidaty, W. Hamidouche, O. Deforges, and P. Philippe, "Compression efficiency of the emerging video coding tools," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2996–3000.

[118] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Nouvelles méthodes de prédiction inter-images pour la compression d'images et de vidéos

Mots clés : Traitement d'image, vision par ordinateur, compression vidéo

Résumé : En raison de la grande disponibilité des dispositifs de capture vidéo et des nouvelles pratiques liées aux réseaux sociaux, ainsi qu'à l'émergence des services en ligne, les images et les vidéos constituent aujourd'hui une partie importante de données transmises sur internet. Les applications de streaming vidéo représentent ainsi plus de 70% de la bande passante totale de l'internet. Des milliards d'images sont déjà stockées dans le cloud et des millions y sont téléchargés chaque jour. Les besoins toujours croissants en streaming et stockage nécessitent donc une amélioration constante des outils de compression d'image et de vidéo. Cette thèse vise à explorer des nouvelles approches pour améliorer les méthodes actuelles de prédiction inter-images. De telles méthodes tirent parti des redondances entre images similaires, et ont été développées à l'origine dans le contexte de la vidéo compression. Dans une première partie, de nouveaux outils de prédiction inter globaux et locaux sont associés pour améliorer l'efficacité des schémas de compression de bases de données d'image. En associant une compensation géométrique et photométrique globale avec une prédiction linéaire locale, des améliorations significatives peuvent être obtenues.

Une seconde approche est ensuite proposée qui introduit un schéma de prédiction inter par régions. La méthode proposée est en mesure d'améliorer les performances de codage par rapport aux solutions existantes en estimant et en compensant les distorsions géométriques et photométriques à une échelle semi locale. Cette approche est ensuite adaptée et validée dans le cadre de la compression vidéo. Des améliorations en réduction de débit sont obtenues, en particulier pour les séquences présentant des mouvements complexes réels tels que des zooms et des rotations. La dernière partie de la thèse se concentre sur l'étude des méthodes d'apprentissage en profondeur dans le cadre de la prédiction inter. Ces dernières années, les réseaux de neurones profonds ont obtenu des résultats impressionnants pour un grand nombre de tâches de vision par ordinateur. Les méthodes basées sur l'apprentissage en profondeur proposées à l'origine pour de l'interpolation d'images sont étudiées ici dans le contexte de la compression vidéo. Des améliorations en terme de performances de codage sont obtenues par rapport aux méthodes d'estimation et de compensation de mouvements traditionnelles. Ces résultats mettent en évidence le fort potentiel de ces architectures profondes dans le domaine de la compression vidéo.

Title Towards novel inter-prediction methods for image and video compression

Keywords : Image Processing, Computer Vision, Video compression

Abstract: Due to the large availability of video cameras and new social media practices, as well as the emergence of cloud services, images and videos constitute today a significant amount of the total data that is transmitted over the internet. Video streaming applications account for more than 70% of the world internet bandwidth. Whereas billions of images are already stored in the cloud and millions are uploaded every day. The ever growing streaming and storage requirements of these media require the constant improvements of image and video coding tools. This thesis aims at exploring novel approaches for improving current inter-prediction methods. Such methods leverage redundancies between similar frames, and were originally developed in the context of video compression. In a first approach, novel global and local inter-prediction tools are associated to improve the efficiency of image sets compression schemes based on video codecs. By leveraging a global geometric and photometric compensation with a locally linear prediction, significant improvements can be obtained.

A second approach is then proposed which introduces a region-based inter-prediction scheme. The proposed method is able to improve the coding performances compared to existing solutions by estimating and compensating geometric and photometric distortions on a semi-local level. This approach is then adapted and validated in the context of video compression. Bit-rate improvements are obtained, especially for sequences displaying complex real-world motions such as zooms and rotations. The last part of the thesis focuses on deep learning approaches for inter-prediction. Deep neural networks have shown striking results for a large number of computer vision tasks over the last years. Deep learning based methods proposed for frame interpolation applications are studied here in the context of video compression. Coding performance improvements over traditional motion estimation and compensation methods highlight the potential of these deep architectures.