DISSERTAÇÃO DE MESTRADO

MOTION COMPENSATION WITH MINIMAL RESIDUE DISPERSION MATCHING CRITERIA

Gabriel Lemes Silva Luciano de Oliveira

Brasília, fevereiro de 2016

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

MOTION COMPENSATION WITH MINIMAL RESIDUE DISPERSION MATCHING CRITERIA

Autor

Gabriel Lemes Silva Luciano de Oliveira

Prof. Ricardo Lopes de Queiroz, Ph.D.

Orientador

Prof. Eduardo Peixoto Fernandes da Silva, Ph.D.

Coorientador

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

MOTION COMPENSATION WITH MINIMAL RESIDUE DISPERSION MATCHING CRITERIA

Gabriel Lemes Silva Luciano de Oliveira

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Ricardo Lopes de Queiroz, CiC/UnB _____
Orientador

Francisco Assis de Oliveira Nascimento, ENE/UnB _____
Examinador interno

Bruno Luiggi Macchiavello Espinoza, CiC/UnB _____
Examinador externo

Camilo Chang Dórea, CiC/UnB _____
Suplente

Dedicatória

Este trabalho é dedicado aos meus pais, Nívea Lemes da Silva e Estanislau Luciano de Oliveira, cujo apoio segue incondicional. Este trabalho é resultado do esforço de vocês.

Gabriel Lemes Silva Luciano de Oliveira

Agradecimentos

Gostaria de agradecer aos meus orientadores, Prof. Ricardo Lopes de Queiroz e Prof. Eduardo Peixoto, pelo acompanhamento atento ao longo deste trabalho e pela contínua motivação. Quero agradecer também à minha namorada, Maria Karolina Beckman Pires, e ao meu amigo, Laércio Martins Oliveira Silva, pela companhia e pelo suporte emocional ao longo desses anos. Agradeço ainda ao amigo e companheiro de jornada, Gustavo Luiz Sandri, pela grande ajuda em incontáveis detalhes. Por fim, quero registrar também a minha gratidão ao Prof. João Luiz Azevedo de Carvalho, cujo voto de confiança me rendeu esta oportunidade. A todos vocês, muito obrigado.

Gabriel Lemes Silva Luciano de Oliveira

RESUMO

Com a crescente demanda por serviços de vídeo, técnicas de compressão de vídeo tornaram-se uma tecnologia de importância central para os sistemas de comunicação modernos. Padrões para codificação de vídeo foram criados pela indústria, permitindo a integração entre esses serviços e os mais diversos dispositivos para acessá-los. A quase totalidade desses padrões adota um modelo de codificação híbrida, que combina métodos de codificação diferencial e de codificação por transformadas, utilizando a *compensação de movimento por blocos* (CMB) como técnica central na etapa de predição. O método CMB tornou-se a mais importante técnica para explorar a forte redundância temporal típica da maioria das sequências de vídeo. De fato, muito do aprimoramento em termos de eficiência na codificação de vídeo observado nas últimas duas décadas pode ser atribuído a refinamentos incrementais na técnica de CMB. Neste trabalho, apresentamos um novo refinamento a essa técnica.

Uma questão central à abordagem de CMB é a *estimação de movimento* (EM), ou seja, a seleção de *vetores de movimento* (VM) apropriados. Padrões de codificação tendem a regular estritamente a sintaxe de codificação e os processos de decodificação para VM's e informação de resíduo, mas o algoritmo de EM em si é deixado a critério dos projetistas do codec. No entanto, embora praticamente qualquer critério de seleção permita uma decodificação correta, uma seleção de VM criteriosa é vital para a eficiência global do codec, garantindo ao codificador uma vantagem competitiva no mercado. A maioria do algoritmos de EM baseia-se na minimização de uma função de custo para os blocos candidatos a predição para um dado bloco alvo, geralmente a *soma das diferenças absolutas* (SDA) ou a *soma das diferenças quadradas* (SDQ). A minimização de qualquer uma dessas funções de custo selecionará a predição que resulta no *menor* resíduo, cada uma em um sentido diferente porém bem definido.

Neste trabalho, mostramos que a predição de mínima *dispersão* de resíduo é frequentemente mais eficiente que a tradicional predição com resíduo de mínimo tamanho. Como prova de conceito, propomos o *algoritmo de duplo critério de correspondência* (ADCC), um algoritmo simples em dois estágios para explorar ambos esses critérios de seleção em turnos. Estágios de minimização de dispersão e de minimização de tamanho são executadas independentemente. O codificador então compara o desempenho dessas predições em termos da relação taxa-distorção e efetivamente codifica somente a mais eficiente. Para o estágio de minimização de dispersão do ADCC, propomos ainda o *desvio absoluto total com relação à média* (DATM) como a medida de dispersão a ser minimizada no processo de EM. A tradicional SDA é utilizada como a função de custo para EM no estágio de minimização de tamanho. O ADCC com SDA/DATM foi implementado em uma versão modificada do software de referência JM para o amplamente difundido padrão H.264/AVC de codificação. Absoluta compatibilidade a esse padrão foi mantida, de forma que nenhuma modificação foi necessária no lado do decodificador. Os resultados mostram aprimoramentos significativos com relação ao codificador H.264/AVC não modificado.

ABSTRACT

With the ever growing demand for video services, video compression techniques have become a technology of central importance for communication systems. Industry standards for video coding have emerged, allowing the integration between these services and the most diverse devices. The almost entirety of these standards adopt a hybrid coding model combining differential and transform coding methods, with *block-based motion compensation* (BMC) at the core of its prediction step. The BMC method have become the single most important technique to exploit the strong temporal redundancy typical of most video sequences. In fact, much of the improvements in video coding efficiency over the past two decades can be attributed to incremental refinements to the BMC technique. In this work, we propose another such refinement.

A key issue to the BMC framework is *motion estimation* (ME), i.e., the selection of appropriate *motion vectors* (MV). Coding standards tend to strictly regulate the coding syntax and decoding processes for MV's and residual information, but the ME algorithm itself is left at the discretion of the codec designers. However, though virtually any MV selection criterion will allow for correct decoding, judicious MV selection is critical to the overall codec performance, providing the encoder with a competitive edge in the market. Most ME algorithms rely on the minimization of a cost function for the candidate prediction blocks given a target block, usually the *sum of absolute differences* (SAD) or the *sum of squared differences* (SSD). The minimization of any of these cost functions will select the prediction that results in the *smallest* residual, each in a different but well defined sense.

In this work, we show that the prediction of minimal residue *dispersion* is frequently more efficient than the usual prediction of minimal residue size. As proof of concept, we propose the *double matching criterion algorithm* (DMCA), a simple two-pass algorithm to exploit both of these MV selection criteria in turns. Dispersion minimizing and size minimizing predictions are carried out independently. The encoder then compares these predictions in terms of rate-distortion performance and outputs only the most efficient one. For the dispersion minimizing pass of the DMCA, we also propose the *total absolute deviation from the mean* (TADM) as the measure of residue dispersion to be minimized in ME. The usual SAD is used as the ME cost function in the size minimizing pass. The DMCA with SAD/TADM was implemented in a modified version of the JM reference software encoder for the widely popular H.264/AVC coding standard. Absolute compliance to the standard was maintained, so that no modifications on the decoder side were necessary. Results show significant improvements over the unmodified H.264/AVC encoder.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BD-rate	Bjontegaard Delta Rate
BMC	Block-based Motion Compensation
DCT	Discrete Cosine Transform
DMCA	Double Matching Criterion Algorithm
DPCM	Differential Pulse Code Modulation
EIP	Enhanced Inter Prediction
FPS	Frames per Second
FS	Full Search
ME	Motion Estimation
MQPT	Multiple QP Testing
MV	Motion Vector
PSNR	Peak Signal to Noise Ratio
R-D	Rate-Distortion
RDO	Rate-Distortion Optimization
RDOQ	Rate-Distortion Optimized Quantization
SAD	Sum of Absolute Differences
SI	Spatial Perceptual Information
SSD	Sum of Squared Differences
ST	Shifting Transformation
TADM	Total Absolute Deviation from the Mean
TI	Temporal Perceptual Information

Chapter 1

Introduction

The ever increasing demand for video data calls for higher and higher video coding efficiency. Video compression relies on a combination of many techniques devised to remove the inherent redundancy typically found in video signals. Of these techniques, motion compensation usually stands out as one of the most important. After a brief review on video coding concepts, we propose a new approach to motion compensation. This chapter gives a quick overview on the subject and provides a road map to this manuscript.

1.1 Video Coding

Transmission or storing of raw video data is mostly infeasible for many applications. The bandwidth or storing devices requirements for many services we take for granted today, such as YouTube™or film distribution through Blu-ray Discs™, would be quite simply prohibitive if raw video data was to be assumed [1]. More than a simple improvement, video compression techniques are an enabling technology.

In addition to the continuous development of increasingly efficient video compression techniques, the development of industry standards has also been critical to the widespread adoption of video technology. The importance of standardization cannot be overemphasized. It allows for the interoperability of a multitude of devices with different resources or from different manufactures, enabling the processing, transmission, and displaying of video data from a wide range of possible sources by a wide range of potential users. Of course, standards allow some flexibility to accommodate competition and are continuously revised to avoid stifling of compression technologies. Nonetheless, conformance to well established standards can be decisive for the success of new techniques since it dictates the cost of adapting already deployed equipment.

Video signals, despite its own idiosyncrasies, are most easily understood as time sequences of correlated pictures or *frames*. As such, many lossy and lossless image compression techniques present themselves as viable tools to video compression. In fact, for example, the block-based DCT transform coding framework of the successful JPEG [2] standard for image coding is also found at the core of many modern video coding standards [1, 3]. However, those techniques by

themselves are not enough to provide the compression rates needed in most applications. The key to high quality video at accessible bit-rates lies in the high correlation present in most video signals between frames in the time dimension. Not only does this correlation allows highly efficient inter-frame predictions for a differential coding framework, the way the human brain exploits this temporal correlation to process visual information also allows us to get away with discarding *a lot* of the numerical information, without hurting the video quality perceived by the end user [3]. Video data is intended, after all, to be analysed or enjoyed by human consumers in most applications.

Several techniques have been developed over the years to exploit the temporal redundancy so characteristic of video signal, of which *block-based motion compensation* (BMC) is arguably the most successful [4, 5]. Most modern video coding standards offer support for BMC. In fact, most of these standards adopt a *hybrid* coding model with BMC at its core. In this hybrid framework, a prediction of each frame to be coded is formed with BMC, based on previously coded frames. The resulting *residual*, the difference between the target frame and its prediction, is then transform coded with a block DCT transform. Much of the improvements in video compression efficiency over the past two decades can be directly attributed to successive refinements to the BMC technique. It is in this context that our own work is inserted. We propose a new such refinement.

To form its predictions, BMC starts by dividing each target frame into several blocks of fixed size. For each target block, a previously coded frame is then searched for a matching block to serve as its prediction. This matching operation itself is known as *motion estimation* (ME). Finally, the encoder outputs the *motion vector* (MV) for the selected prediction block, its relative displacement with respect to the target block. The resulting residual block is used to compose the residual frame, which will be subsequently transform coded. At the decoder side, the frames used for prediction will have already been decoded by the time the MV's for the next frame is received. These MV's can then be used to reconstruct the predicted frame. Also in possession of the residual frame, the decoder can then recover the intended target frame.

Observe that the whole decoding operation is entirely transparent to the actual MV selection process. That is, given a set of MV's, if they are encoded together with an appropriately formed residual frame, the decoder can successfully recover the intended target frame even if the MV's are selected at random. Evidently, however, judicious MV selection leads to MV sets and residual frames that can be more efficiently encoded, providing the encoder with a competitive edge on the market. Therefore, most video coding standard tend to regulate only the encoding syntax and the decoding processes for MV's and residual frames. The actual ME algorithms are let at the discretion of the designers, thus fomenting competition and innovation all while promoting the desired interoperability in compliant systems.

There are two key issues to the ME operation, namely, the definition of a *match* and the search algorithm. A match is defined in terms of a *matching criterion*, which is usually the minimization of a cost function, also referred to as the *distortion measure* between the prediction and target blocks. The search algorithm then tests a number of candidate blocks and selects the one that minimizes the predefined distortion measure to be the actual prediction block to the given target. The most straight forward strategy is a *full search* (FS) algorithm, wherein every single one of all

possible candidate blocks within a given *search area* is tested with respect to each given target block in each target frame in terms of the selected distortion measure [4, 7]. While the FS algorithm is guaranteed to reach the smaller residue within the constraints of the search area, it might be too computationally expensive for many applications, especially in situations requiring real time coding. The are several algorithms designed to offer different levels of trade-off between prediction optimality and computational requirements [4, 8]. Our focus within this work, however, lies on the distortion measures and matching criteria.

1.2 Objectives: Minimal Dispersion Matching Criteria for ME

Some of the most popular matching criteria are the minimization of the *sum of squared differences* (SSD) and the minimization of the *sum of absolute differences* (SAD), possibly weighting the cost for encoding the appropriate motion vector [6]. The SSD and the SAD are both functions of the residual alone, i.e., they both depend only on the *difference* between the prediction and target block, but not on their actual values individually. Both of these cost functions are directly related to well defined notions of *distance*, the L_2 and the L_1 norms, respectively. Therefore, the minimization of any of them will result in a residual that is the smallest possible, each in a different but well defined sense.

In this work, we argue that, instead of always choosing the prediction of smallest residual, it is sometimes more efficient to chose the prediction that results in the residual most *concentrated* around a central value, even if it may result in a large residual. A value around which a collection of values tend to cluster is known as a *central tendency* of those sample values. Common measures of central tendency include the mean, the median, and the mode. The spread of these values around their central tendency is known as their *dispersion*. The most common measure of dispersion is the mean squared deviation from the mean, also know as the *variance*, but several others can be defined. In other words, in this work, we argue that the prediction of minimal residue dispersion is sometimes more efficient than the usual prediction of smallest residue.

There is a strong precedent for ME with minimal residue dispersion. However, it has gone largely unnoticed in these terms, since it was given a very different interpretation. In 2012, Blasi et al. proposed the *enhanced inter-prediction* (EIP) method to improve the BMC approach [9]. Their method consisted in transforming the candidate blocks with an invertible parametric transformation to better match the target block, only then comparing the candidate and target blocks in terms of the SAD or in terms of the SSD. For each candidate, the parameters of the transformation are optimized to that end. The parameters used for the winning block are then sent along with the respective residue and motion information to the decoder over the bitstream, so that it can invert the transformation. The premise behind EIP is that the extra bits spent coding the parameters of the transformation are offset by a smaller residue, which would require fewer bits to code.

As a proof of concept for the EIP, Blasi et al. also proposed the *shifting transformation* (ST) [9]. It consisted in a single parameter transformation which itself consisted in uniformly adding a single constant to each block. For each block, the constant was optimized in the sense of minimizing the

resulting residual after the comparison with the target block. They also devised an algorithm to compute the optimal shift if the SAD is used as a measure of the residual and provided a closed solution to the problem if the SSD is used instead. Their implementation of EIP with ST in the H.264/AVC standard using the JM reference software [10] with the SAD metric showed significant gains over the base encoder [9]. In our work, however, rather than in its precise value, we are more interested in a particular interpretation of the optimal shift.

In theory, a very wide range of invertible parametric transformations would be suitable for the EIP approach. However, for the EIP to be effective, the residuals must frequently be smaller enough to compensate for the cost of encoding the optimal parameters for *every* coded block. Actually, then, it is not immediately obvious if there is *any* transform suitable for the EIP approach at all. Taking that into account, it is rather remarkable that the shifting transformation can improve coding efficiency as much as shown in experiments [9]. Therefore, an insight on *why* EIP with ST works might provide useful insight on the general matching criterion definition problem.

We show that, although it keeps the SAD or the SSD as a measure of distortion between the transformed candidate block and the target block, the EIP approach changes the matching criterion on its essence. In fact, in the case of the EIP with ST, the distortion measure itself is fundamentally changed. The minimization of the SAD or of the SSD for the transformed candidate blocks, given the target, results in the smallest residual possible. In general, it is always smaller than or equal in size to the usual residual, since the usual prediction blocks themselves are also accounted for with a zero shift parameter. That is, in terms of the *transformed* candidate blocks, it would seem that nothing changed in the matching criterion, except that more candidate blocks are tested due to the various possible values of the shifting parameter. In terms of the *original* candidate block, however, the shifted SAD and SSD are no longer measures of size in any sense. They do still have an interesting interpretation, though.

What we show in this work is that, irrespectively of whether the SAD or the SSD is used, the optimal shift parameter is a function of the residual alone, as are the SAD and the SSD themselves. Therefore, instead of testing a larger set of candidate blocks in terms of the usual distortion functions, the EIP with ST effectively tests the *same* set of candidates in terms of a completely different distortion function. Although not immediately clear in the original work on the EIP with ST by Blasi et al., the optimum shift parameter in the case of SAD distortion measure is simply the negative *median* of the residual block, as we show later in this work. The SAD of the residual block shifted by its negative median is simply the absolute deviation of the residual from its median value, a measure of *dispersion*, much like the variance. In the case of the SSD distortion measure, as shown already by Blasi et al. in their original work, the optimum shift is simply the negative *mean* of the residual. The SSD of the residual block displaced by its mean is also another measure of dispersion, actually proportional to the variance itself. In other words, in the EIP with ST approach, the matching criterion is changed so that, instead of searching for the prediction of smallest residue, the encoder actually searches for the prediction of minimal residue dispersion.

Our goal is to show the effectiveness of motion estimation with minimal residue dispersion

matching criteria in the BMC framework. Although we believe that the EIP with ST already lends strong testimony to that end, we provide further proof of concept while overcoming the most important shortcoming of the EIP approach, namely, the need to code and transmit the optimum shift parameter separately, which renders it non-compliant to established video coding standards. We devise a two-pass motion estimation algorithm combining both dispersion measures and distance measures. It is implemented it in the widely popular H.264/AVC standard, with extensive testing showing significant improvement in coding performance. The generated bitstream is made fully compliant to the standard, meaning that the proposed technique can be implemented in an H.264/AVC coding system without the need to upgrade or replace decoders.

1.3 Manuscript Organization

The remaining of this work is organized as follows. In Chapter 2, video coding concepts and techniques, including the EIP, are discussed in greater detail. In Chapter 3, we present our proposed algorithm and its development with heuristic considerations over the EIP with ST leading to dispersion measures as matching criteria for ME. In Chapter 4, we show some comparative results demonstrating the gains in coding efficiency achieved with the proposed algorithm. Finally, we conclude our work in Chapter 5.

Chapter 2

Video Coding

Video data can be viewed as a temporal sequence of images, which allows for a wide range of image compression techniques and concepts to be adapted to video compression. In most video sequences, however, these images are highly correlated to their temporal neighbours. Techniques developed to exploit this temporal correlation between frames are collectively known as *inter frame coding*. In this chapter, we briefly review some of the most important concepts and techniques in video compression, with particular attention to one such inter frame coding technique known as *block-based motion compensation*.

2.1 Video Coding Concepts

Digital video sequences may arise form several processes such as computer animations or the digitalization of natural or "real world" scenes. Either way, a digital video signal can be viewed as a temporal sequence of still images, known as *frames* in this context. Each frame is a rectangular array of color samples known as *pixels*, as illustrated in Figure 2.1. If each frame consists of N lines by M columns of pixels, it is said the video sequence has $N \times M$ *resolution*. These frames are supposed to be displayed at a fixed rate known as the *temporal resolution* or simply the *frame rate*, measured in *frames per second* (fps), to create the illusion of motion. The *duration* of a video sequence is the amount of time required to reproduce the video sequence at the required frame rate.

When we speak of a video signal or sequence, we are usually referring to a representation of the signal suitable for immediate display, that is, a 3-dimensional array of pixels, whose colors are coded with a fixed amount of bits known as *depth*. Finally, the *size* of a video sequence refers to the amount of bits required to represent it. For example, 10 seconds of a 128×96 resolution video sequence at 30 fps with 24 bits per pixel for color coding has size $10 \times 30 \times 128 \times 96 \times 24 = 88473600$ bits, or about 11MB. We might also refer to the bit *rate* of the signal, which is usually the amount of bits per second required to represent the sequence, but might also refer to the amount of bits per frame or per pixel.

The ultimate goal of video compression is the efficient representation of a digital video sequence.

A video compression system is actually composed by two complementary systems, an *encoder* and a *decoder*, also collectively referred to as a *CODEC pair*, or simply a codec. The input to the encoder is a digital video sequence $\mathcal{X}$. It produces a representation $\mathcal{Y}$ of this video sequence, better suitable for storage or transmission, known as the *bitstream*. The bitstream is the input to the decoder, whose output is a reconstruction $\hat{\mathcal{X}}$ of the original sequence. The process is illustrated in Figure 2.2.

Video compression is usually *lossy*, which means we usually allow for $\hat{\mathcal{X}} \neq \mathcal{X}$ [12, 3, 1]. Therefore, the performance of a video compression system is measured *both* by the bit savings of the representation $\mathcal{Y}$ over the representation $\mathcal{X}$ *and* by how well the reconstruction $\hat{\mathcal{X}}$ approximates the original sequence $\mathcal{X}$.

Assessment of compression performance is straightforward in terms of $R_{\mathcal{Y}}$ and $R_{\mathcal{X}}$, the rates of the compressed and uncompressed signals, respectively. The quality of the reconstruction, however, is quite an elusive concept, since video sequences are mostly intended to be ultimately appreciated or analysed by human viewers [3]. Ideally, a measure of quality should convey this rather subjective notion of quality in the minds of the intended audience. Such a measure is very hard to devise, so we usually get by defining a more mathematically tractable measure of *distortion* $D = f(\hat{\mathcal{X}} - \mathcal{X})$, a function of the difference between the reconstructed and original signals, such as the signal-to-noise ratio or the mean square error [3]. It is assumed that the smaller the distortion the higher the quality of the reconstruction.

The performance of the compression system for a given sequence $\mathcal{X}$ is then given by the achievable rate $R_{\mathcal{Y}}$ at a given distortion D or, alternatively, by the achievable distortion $D_{\hat{\mathcal{X}}}$ at a given

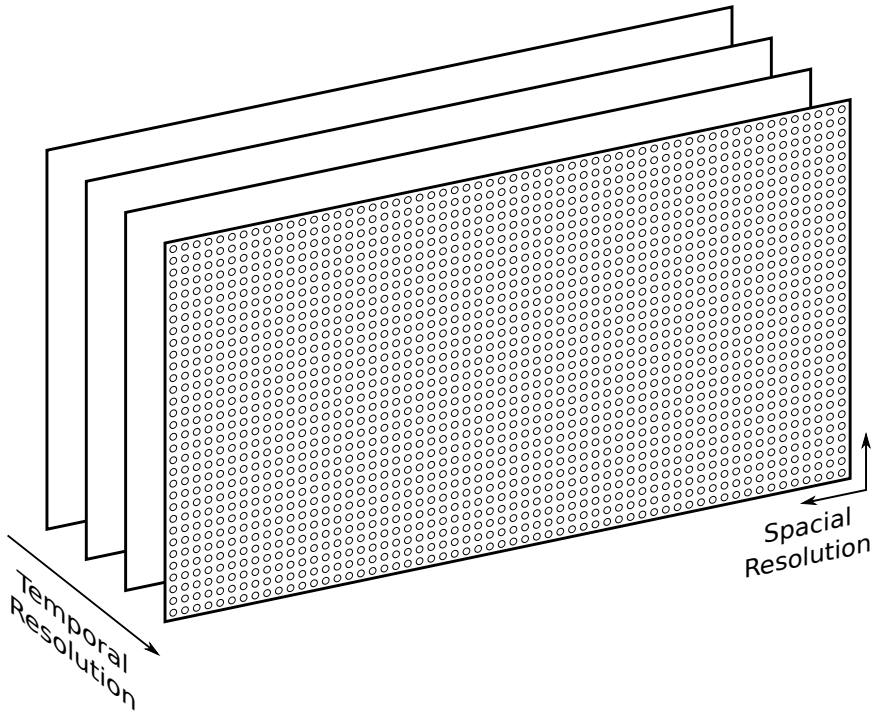

Figure 2.1: Video signals sampling scheme.

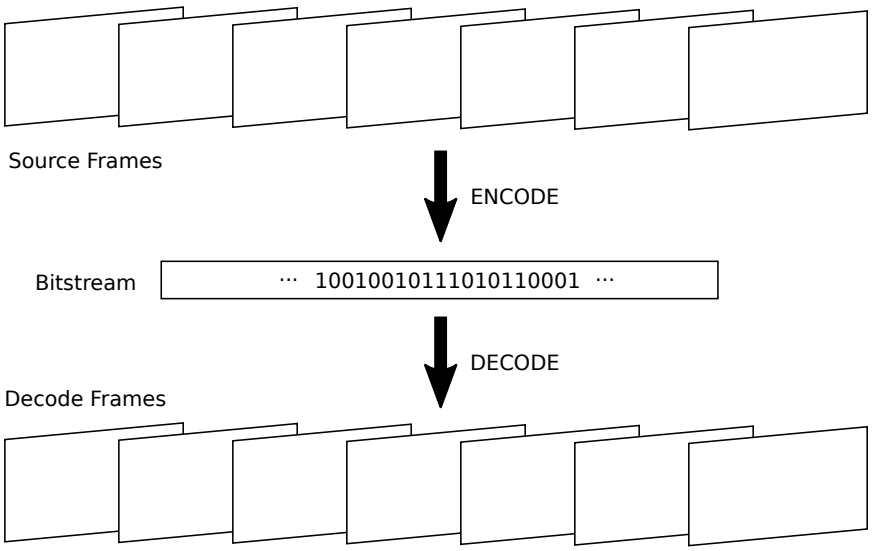

Figure 2.2: Encoding/decoding process.

rate R, as illustrated in Figure 2.3. The fundamental limits on the performances attainable in this sense by any lossy compression scheme are given by the rate-distortion theory [13, 3], a sub-field of information theory. Given a source S and its probability model, we define the *rate-distortion function* $R_S(D)$ as the lowest possible rate to describe it with distortion at most D. The *distortion-rate function* $D_S(R)$ is analogously defined. In theory, $R_S(D)$ or $D_S(R)$ are the milestones to which $R_{\mathcal{Y}}(D)$ or $D_{\hat{\mathcal{X}}}(R)$ should be compared. In practice, however, it is very hard to devise satisfactory probability models to complex sources such as video and, even given one, R_S and D_S are notoriously difficult to evaluate for all but a few simple probability models. Nonetheless, rate-distortion theory does provide insight for practical lossy coding.

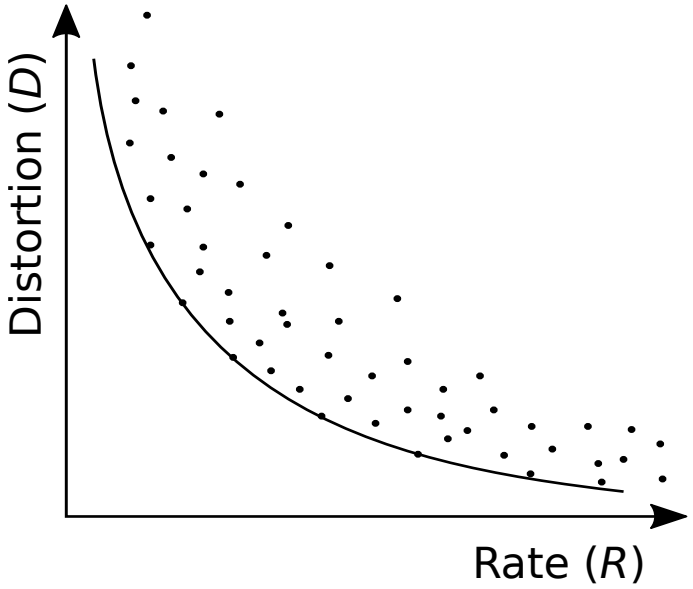

Figure 2.3: Rate-distortion operations points points for fixed sequence and fixed codec at different configuration options. The convex hull delineated in the plot indicate the achievable rate-distortion performance for this given codec-sequence pair.

2.2 Video Coding Techniques

As with any data compression scheme, video compression is achieved by removing the *redundancy* inherent to video data [3]. *Statistical* redundancy can be removed or mitigated with *lossless* entropy coding methods [13, 12]. *Perceptual* or *subjective* redundancy can be removed by *lossy* compression methods [3, 1]. These methods may incur in some data loss since they invariably employ one form or another of *quantization*, an irreversible operation. However, far greater compression is possible with lossy compression. For sound or image compression, for example, there are several clever methods exploiting psychoacoustic or psycho-visual phenomena to selectively discard data to which the human brain is less sensitive, allowing for a very good trade-off between compression and quality perceived by the end user [3]. In fact, for video coding at a fixed bit rate, if the sampling scheme is allowed to change, it is possible that a *greater* quality is perceived by the end user with lossy compression methods, since it allows for higher resolutions and frame rates than those of a losslessly coded sequence at the same bit rate [1].

Structure inherent to most signals of interest, which manifest itself in the form of highly correlated samples, can be exploited to improve lossless entropy coding. For most signals, however, this structure is very difficult to grasp in a probability model, required for direct applications of entropy coding methods. Instead, it is usually preferable to produce a new, less correlated representation of the data, in which its structure is described in a way that makes it amenable to the usual entropy coding. *Differential coding* and *transform coding* are two widely popular classes of methods to devise such representations [3]. Furthermore, the structure revealed by these representations can often enable a more efficient trade-off between rate and distortion in lossy coding methods. Video codecs employ methods in both classes to remove both *temporal* correlation and *spatial* correlation, typical of video data.

Differential coding, often also referred to as *differential pulse code modulation* or DPCM [12], works by producing a *prediction* of a value or set of values to be coded. The encoder then forms a *residual*, the difference between the values to be coded and the prediction. The residual is then coded, together with any information necessary for the decoder to reproduce the same prediction. In face of lossy encoding, since the prediction formed at the decoder uses only past decoded values to form the prediction, the encoder must implement a decoding loop itself so that it can produce its predictions in synchrony with the decoder. Otherwise, the lost synchrony at the decoder produces a cumulative error effect known as *drifting* [3].

Transform coding works by encoding a transformed representation of the values to be coded. The transformation must be invertible, so that the decoder can recover the original values, or an approximation of the original values in the case of lossy compression. For image compression, for example, the selected transform might operate in the whole image, as in the case of the wavelet transform used in JPEG2000 [14], or in blocks of pixels, as in the case of the block DCT used in JPEG [2].

2.2.1 Intra Coding and Inter Coding

Since we can view a video signal as a temporal sequence of still images, a host of image compression techniques become readily available as tools for video compression as well. In fact, some compression can be obtained by representing a video sequence as a series of independently compressed still images, removing redundancy due to spacial correlation between samples within each frame. This approach is known as *intra* coding. Intra coding can only achieve a limited amount of compression, however, since it overlooks a great deal of correlation typically present in the temporal domain in most video signals. Techniques that exploit this temporal correlation, taking advantage of information present in preciously coded frames to improve the compression of frames to be coded are collectively know as *inter* coding.

Both differential coding and transform coding are suitable techniques for either intra coding or inter coding. Prediction formed in the intra coding framework is also known as *intra-prediction*, while prediction techniques for inter coding are also known as *inter-prediction*. We refer to a frame coded exclusively with intra coding techniques as an *intraframe*, and to a frame that uses inter-prediction techniques, exclusively or not, as an *interframe*.

Most video codecs today employ both inter and intra coding techniques. Though inter coding allows for greater compression, the periodic insertion of intraframes in the stream can add some benefits. For example, it can avoid to some extent the propagation of decoding errors in time and it also allows the access to the video content at random points in time without the need to decode the entire sequence up to the desired point. Moreover, even in interframes, it might be more efficient to code some areas with intra coding techniques.

2.2.2 Block Motion Compensation

In this work, our focus lies in inter coding. In particular, we focus in an inter-prediction technique known as *block motion compensation* (BMC) [1]. In the basic BMC approach, a frame to be coded is divided in non-overlapping blocks of fixed size. For each block, a prediction is formed by selecting a matching block in the previous frame. Note that the prediction need not conform to the grid induced by the fixed size blocking. The encoder then forms a *residual* block and encodes it together with the offset between the position of the *target* block to be encoded and the position of its prediction. This offset is known as a *motion vector* (MV). The matching operation itself, or MV selection, is known as *motion estimation* (ME).

Some commonly used extensions to this method include optional block splitting for more locally adaptive predictions [15], non-integer MV's to allow for closer matches from interpolated frames [16], and multiple reference frames to better model long term correlations in time [17]. We cover BMC in greater detail later in this chapter, including the precise definition of a "match" and some common algorithms for ME.

2.2.3 Hybrid Coding

Most major video codecs developed since the early 90's share a basic model known as *hybrid DPCM/DCT coding* for inter coding. This model consists in a predictive step with BMC to form a residual frame, which is then transform-coded with a block DCT. The motion compensated prediction step promotes decorrelation in the time domain. The subsequent DCT step, besides promoting further decorrelation in the spacial domain, also allows for more efficient quantization, taking advantage of the energy compacting properties of the DCT as well as the fact that video signals are usually intended to be consumed by human viewers. Since it is known that the human visual system is less sensitive to high frequency information [3], higher frequency AC coefficients of the residue DCT can usually be more aggressively quantized without great degradation of perceived quality to the end user [1], resulting in a better trade-off between rate and quality. The quantized residue DCT coefficients are then entropy coded and finally written in the bitstream. Figures 2.4 and 2.5 summarizes the hybrid encoding and decoding process, respectively. Observe the decoding loop in the encoder size to avoid drifting.

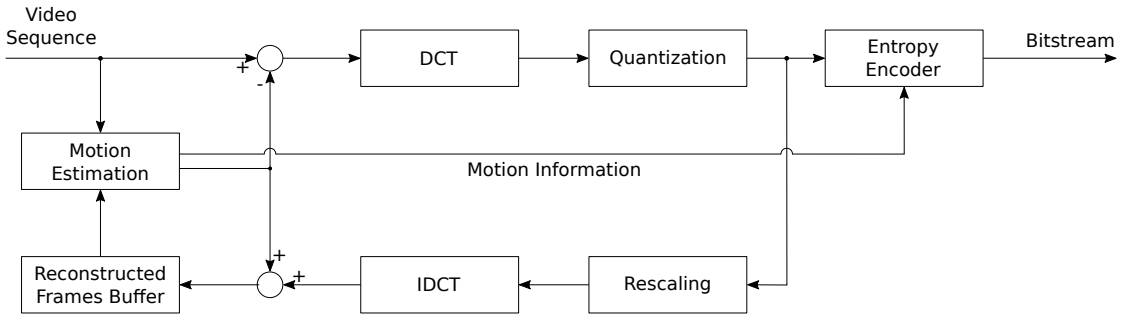

Figure 2.4: Hybrid encoder.

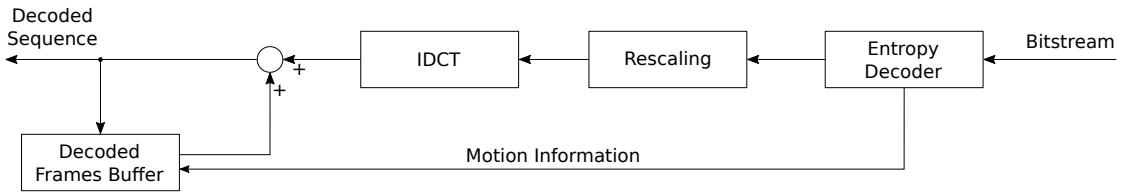

Figure 2.5: Hybrid decoder.

This hybrid coding scheme is not exclusive to inter coding. In fact, some codecs also employ hybrid coding for intra coding [11]. In this case, intra-prediction techniques are used for the differential step.

2.2.4 Rate Distortion Optimization

Video signals vary greatly in the form of its content. Not only between different sequences but also within particular sequences themselves or even in a single frame. Clearly, no single coding technique can efficiently compress general video sequences. Instead, video codecs usually equip the encoder with an arsenal of coding techniques so that it can locally adapt to varying spatio-

temporal characteristics such as texture, movement, and variations in illumination conditions. The bitstream is then formatted so that these local adaptations can be signalled to the decoder, which usually lacks the information or the computational power to reliably infer them.

For example, for each block in an interframe, an encoder might be able to chose between intra prediction or inter prediction. In the first case, it must then decide between several methods usually available for intra prediction. In the second case, several forms of splitting of this fundamental block might be allowed for a finer grained motion estimation. Some times, for example, instead of sending one MV and a 16×16 residual block, it might be more efficient to send four MV's with four 8×8 residual blocks. There is a clear trade-off between the extra bits needed to code the three extra MV's and the smaller residual one expects from this finer motion estimation. In other words, a hybrid encoder can usually select from a range of prediction *modes* for the DPCM step. Local decisions about block transform sizes and quantization scheme for the DCT coding step are also allowed for some codecs.

All these decisions taken at the encoder side must be coded into the bitstream. Once this information is received, the decoder can readily reconstruct each block to the desired approximation. At the encoder, however, the issue of efficiently making these decisions is a complicated and fundamentally important one. A sensible way to approach this decision problem is through *rate-distortion optimization*.

As stated before, the best performance achievable by any video codec is given by the distortion-rate function. Even though we cannot usually evaluate the distortion-rate function, it makes sense to state the mode decision problem and other related parameter selection problems so as to approach it as close as possible when given a particular codec. So we state the decision problem, or parameter selection problem, in terms of the minimization of the overall distortion D subject to the restriction of the rate R to an overall bit budget R_o:

$$\min D \qquad \text{s.t.} \quad R \leq R_o, \tag{2.1}$$

in which the minimization is carried out over all possible combinations of decisions along the coding process of the entire sequence. In practice, however, not only this minimization is infeasible, it might also be undesired in common scenarios where a frame must be coded without access to future frames or where the rate must be tightly controlled to fit a limited channel capacity at all times, not only on a global average [18]. Instead, the optimization is usually carried out locally for each single decision or a small set of related decision taken together, i.e., most of the time, the optimization is carried out over the possible outcomes of each decision with D and R evaluated only for the limited regions *immediately* affected by that particular decision such as a single block or even a sub-partition of a block [19, 20].

In this local approach, however, problem (2.1) might have no meaningful solution if the constraint $R \leq R_o$ cannot be achieved by the particular decision being locally considered, so we perform the minimization of its Lagrangean R-D cost function J instead [20]:

$$\min J \qquad , \qquad J = D + \lambda R, \tag{2.2}$$

in which λ is known as a *Lagrange multiplier* and the minimization is carried out locally for each decision. Besides being always well defined, the minimization of the cost J also has an interesting interpretation as the joint minimization of D and R, with λ as a design parameter which can also be locally adapted to shift emphasis from distortion minimization to rate minimization or vice versa. Figure 2.6 shows how this trade-off is effected. Furthermore, it is known that when problem (2.1) does have a solution, there is always a value λ_o for λ with which problems (2.1) and (2.2) have the same solution [20, 21].

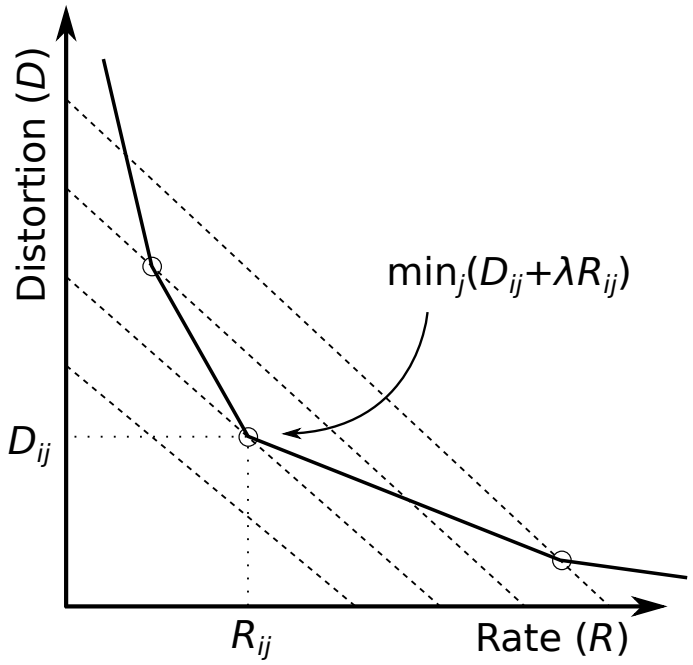

Figure 2.6: The Lagrangean minimization in the rate-distortion space. The dashed lines represent constant-valued Lagrangean functions. Each circled point represents a possible outcome j for decision i. Higher values for the Lagrange multiplier would result in constant-valued Lagrangean lines more inclined to the left, favouring operation points more to the right in the rate-distortion plane, with higher rates and lower distortions. Lower values for the Lagrange multiplier would have the opposite effect.

Rate-distortion optimization (RDO) is usually understood in this local Lagrangean sense. The λ parameter can be heuristically chosen given an user defined parameter of rate or quality [22], or it can be iteratively selected given a target rate. Even then, for some decisions, further simplifications may be needed since the precise calculations of D and R, which involves fully coding and decoding the relevant regions, for each candidate solution in each local decision, can place an unrealistic computational burden on the encoder. It might make sense then to substitute D and R by some approximate estimation thereof [20].

2.3 Standardization and the H.264/AVC Standard

The development of industry standards for video compression and formatting was crucial for the widespread adoption of video technology we witness today. As stated before, video content may arise from a multitude of sources. Furthermore, it may also be intended for displaying in a wide range of devices for different applications. Standardization has enabled this possibility by allowing the interoperability of devices from distinct origins in a video communication system. It accomplishes this goal by strict regulation of how a compressed video bitstream must be formed and decoded.

Coding standards should leave enough room for improvements on coding performance and competition between developers of coding tools. To that end, most standards try to limit its scope as much as possible to the bitstream formatting and decoding process, as shown in Figure 2.7. For example, given the success and widespread adoption of the motion compensation scheme described earlier, most standards provide a strict description of how motion vectors and residual blocks should be written into the bitstream as well as a strict description of how this data will be decoded and used at the decoder to recreate the prediction. That leaves the developers with a lot of freedom at the encoder side to perform motion estimation, including the possibility of selecting the search algorithm and the matching criterion as they see fit for their desired application. A standard can also allow for several prediction modes, providing decoder with means to reproduce these prediction and strictly prescribing how the encoder should write each mode to the bitstream, but will usually not prescribe how mode decisions should be carried out at the encoder side.

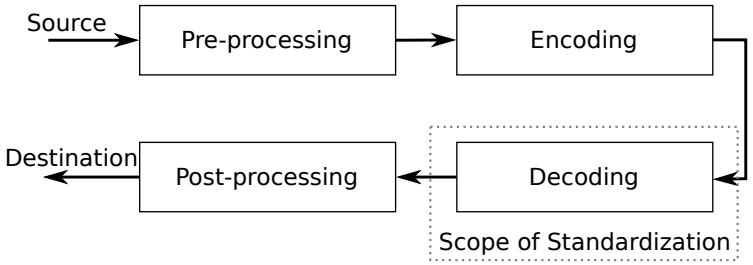

Figure 2.7: Standardization scope.

Effectively, then, standards define a data container and a set of tools available at the decoder. That does, however, limit the overall system performance as well as it *does* limit the freedom at the encoder side to some extent. Back to the motion compensation example, if the standard allows for motion vectors with up to half-pixel accuracy using frame interpolation, the encoder cannot effectively communicate a motion vector with quarter-pixel accuracy to the decoder since there are no provisions for such motion vector in the data container defined by the standard. An encoder is also not allowed to select a prediction mode not provided by the standard. That accounts for the variety of video coding standards in existence today as well as for their continued revisions.

Since the introduction of the H.120 standard through the recent H.265/HEVC, video coding standards have delivered a near halving in video bit rates at roughly ever 10 years over the last 30 years [1]. One of the most successful and still one of the most widely adopted formats is the

H.264/AVC standard which we briefly introduce now. For a historical account on video coding standards development, see [23]. For a more detailed view on the H.264/AVC coding standard, see [24, 25, 11].

The H.264/AVC coding standard, also known as MPEG-4 Part 10, is the result a joint collaboration between ISO/IEC JTC1 Motion Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) [26]. The standard describe an array of coding tools for video encoding and decoding, intended to work in a wide range of applications. In order to manage this variety of scenarios, several *profiles* are prescribed, each defining a subset of these tools which must be supported by a decoder compliant to that profile. In addition, several *levels* are also specified, imposing upper limits on frame size, processing rate and working memory available at the decoder. A particular decoder compliant to a certain combination of profile and level is only required to be able to decode sequences encoded in compliance to combinations of profiles and levels up to its own profile and level combination [11]. In this sense, the H.264/AVC coding standard is actually a family of coding standards.

In order to achieve the flexibility required to meet the needs of a multitude of applications, especially applications over mobile networks and the internet, the H.264/AVC standard defines an hierarchical bitstream syntax. An encoder can then work separately in a *video coding layer* (VCL), designed to efficiently represent video content, and a *network abstraction layer* (NAL), which encapsulates the VCL representation with suitable header information independently from the actual network, relying in external protocols to actually transport or store the bitstream as shown in Figure 2.8.

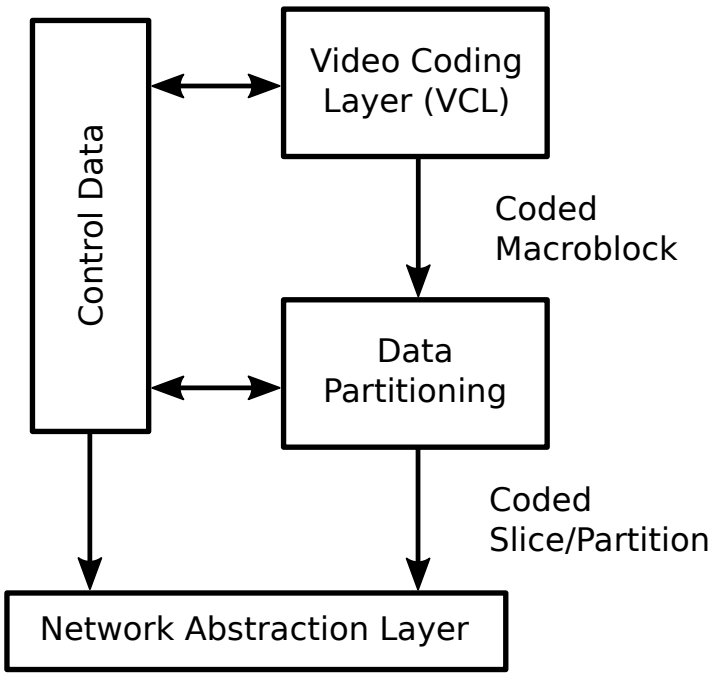

Figure 2.8: Layered encoder operation.

As in most modern codecs, the H.264/AVC VCL design is based in the hybrid block coding scheme described earlier. Each frame is partitioned in fixed size *macroblocks* covering a 16×16

samples square area[1]. Each macroblock is predicted with intra or inter coding techniques and the residual transform-coded with an integer approximation to the DCT. This DCT-like transform operates in 4×4 blocks, with an optional 8×8 transform (not available in some profile-level combinations). A quantization parameter QP, taking 52 integer values from 0 to 51, controls the quantization of the residue transformed values. The quantization step is controlled logarithmically by QP, which provides the primary means of controlling the rate-distortion operation point. Macroblocks are grouped in *slices* for encoding. Each slice either covers an entire frame or non-overlapping regions of a frame, as in Figure 2.9, and each slice is independently coded.

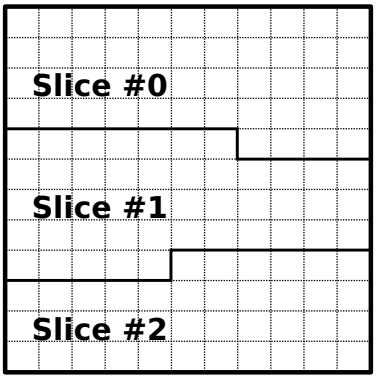

Figure 2.9: Three slices covering a frame.

Slices come in five fundamental types: I slices, P slices, B slices, SP slices, and SI slices. We cover the first three types, see [11] for informations on SP and SI slices.

An *I slice* is a slice in which every macroblock is coded using intra prediction only. There are two basic intra prediction types supported: Intra_4×4 and Intra_16×16. Other modes might be available in some profiles. There are nine prediction modes of the Intra_4×4 type, in each of which a 4×4 prediction block is formed from a set of neighbouring samples in previously coded blocks. The encoder can select one of eight directional prediction modes as in Figure 2.10 or a DC mode. In Intra_16×16 prediction, an entire macroblock is predicted at once with one of four modes: horizontal, vertical, plane or DC.

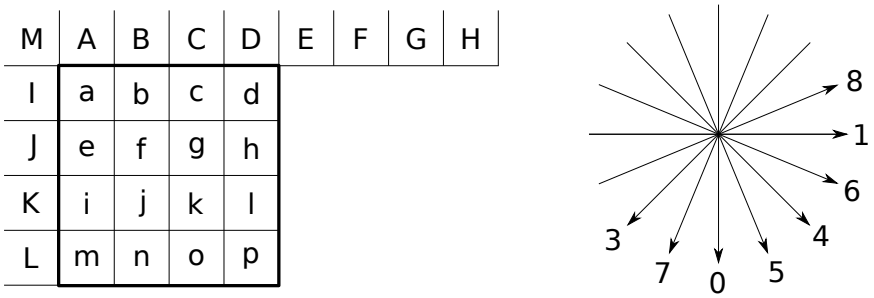

Figure 2.10: The eight 4×4 directional prediction modes. These are complemented by the DC mode, or mode 2, when samples a-p a uniformly predicted from the average from samples A-M.

In a *P slice*, in addition to the intra prediction modes of I slices, a macroblock can also be coded with a motion compensated signal. The syntax allows for multipicture motion compensation,

[1]For *luma* samples, with a corresponding block of *chroma* samples, which is usually smaller due to sub-sampling.

that is, more than one reference frame can be used for motion compensation. Motion vectors in H.264/AVC can have up to quarter integer precision. The filters for half and quarter integer interpolation are also defined by the standard.

BMC can be carried out for an entire 16×16 macroblock or for 16×8, 8×16, 8×8, 8×4, 4×8 or 4×4 blocks, as shown in Figure 2.11. Motion compensation for blocks smaller than 8×8 must all use the same reference picture as the other blocks in its 8×8 region. Figure 2.12 illustrates a possible macroblock partitioning for motion compensation.

Motion vectors are also differentially encoded using either median or direction prediction from neighbouring macroblocks. No prediction takes place in slice boundaries.

A macroblock in a P slice can also be coded in Skip mode, in which no motion or residual information is coded and the reconstructed signal is composed entirely by a prediction formed by the predicted motion vector.

A *B slice*, in addition to the prediction modes allowed for I and P slices, also allows a macroblock to be predicted by the superimposition of *two* motion compensated in a weighted average.

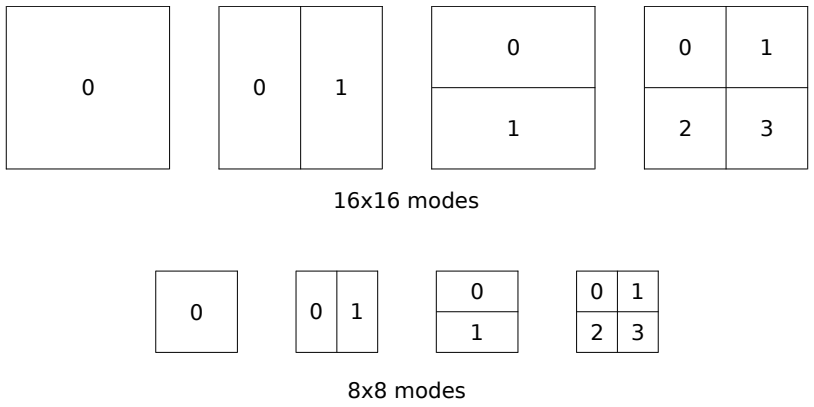

Figure 2.11: Macroblocks partitions for motion compensation.

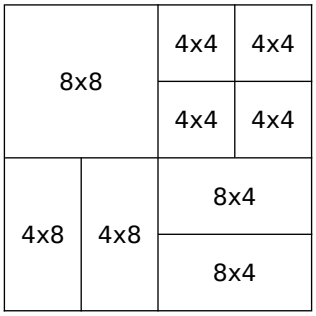

Figure 2.12: A possible macroblock partition.

An H.264 bitstream consists in a series of NAL Units (NALUs), as illustrated in Figure 2.13. The NALU header indicates if it is a *sequence parameter set* (SPS) NALU, a *picture parameter set* (PPS) NALU or a VCL NALU. SPS NALUs contain information that applies to the whole video sequence such as profile, level, resolution and other relevant information to the decoder that are expected to keep constant. PPS NALUs contain more local information relevant for a

group of frames such as the number of slices, the entropy coding mode and other initialization parameters [11]. Each sequence starts with an *instantaneous decoder refresh* (IDR) slice. An IDR slice is an intra coded frame informing the decoder that no future slices requires reference to any slice previous to the IDR slice, allowing the the decoding process to start from there.

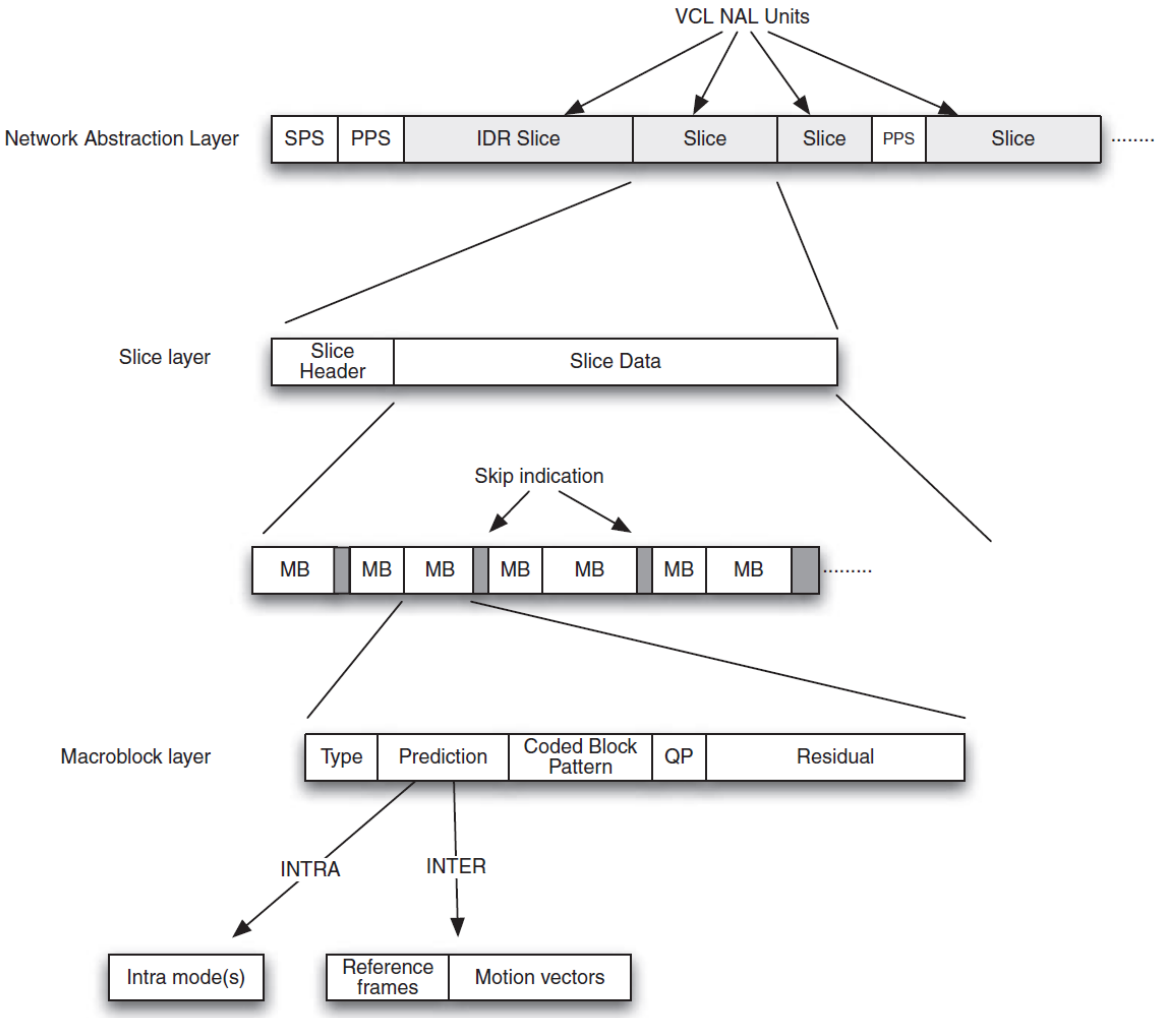

Figure 2.13: Typical H.264/AVC bitstream. Adapted from [11].

2.4 Motion Compensation

The block motion compensation technique briefly introduced in Section 2.2.2 is arguably the most successful method for inter-prediction in video coding [4, 5]. Due to its wide popularity, most video coding standards allows for the effective encoding and decoding of motion compensated sequences. In fact, much of the improvement in video coding efficiency we witnessed in the past two decades derive from the cumulative effects of several small refinements to that basic BMC approach. Our own proposal in this work is another such refinement, so we now proceed to a careful description of the BMC technique.

In the basic BMC framework, a frame to be coded is divided into several blocks of size $n \times m$ pixels, as illustrated in Figure 2.14. Each *target* block T in the frame is sequentially coded as follows. A *search area* around the equivalent position of the target block in a previously coded frame is defined by displacing the equivalent block by $\pm w_x$ and $\pm w_y$ pixels in the horizontal and vertical directions respectively. The encoder then searches within this search area for a *prediction* block P that better matches the target block T. Observe that, if lossy compression is allowed, as is usually the case, the encoder must implement a decoding loop itself and search its predictions within *reconstructed* frames to avoid drifting, as noted in Section 2.2.3. Once the best match is found, the encoder outputs its corresponding *motion vector* (MV), indicated by ν. The search process itself is known as *motion estimation* (ME). The *residual* block $E = P - T$, also known as the *prediction error*, is also sent over the bit-stream. In possession of both ν and E, the decoder can reproduce the prediction P and recover the target block T.

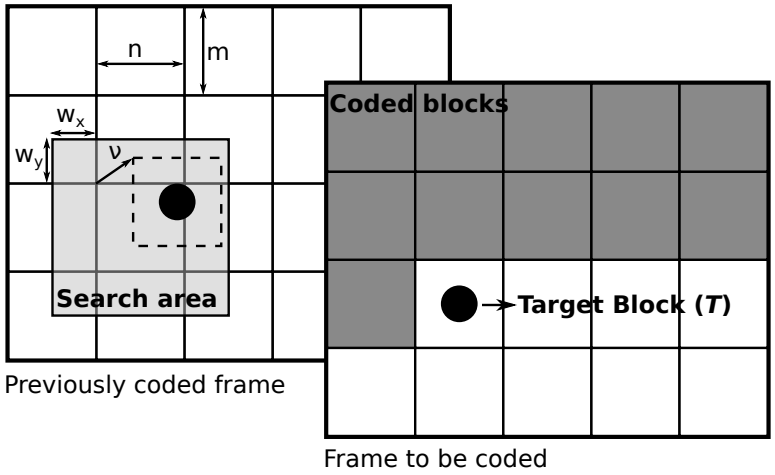

Figure 2.14: Block-based motion compensation.

A better matching for a given target block T is evaluated in terms of a predefined *matching criterion*, usually the minimization of a cost function or *distortion measure* $cost(\cdot, T)$. More precisely, the encoder outputs a motion vector ν_o for a prediction block P_o, along with the corresponding residue $E_o = P_o - T$, if P_o satisfies $cost(P_o, T) \leq cost(P, T)$ for every candidate prediction block P considered. The reasoning behind this scheme is that ν_o and E_o usually require less bits to encode than T itself.

Underlying this BMC prediction approach there is a 2-dimensional rigid body translational motion model. Heuristically, it is expected that a target block and its respective prediction block both correspond to the same region of the same object in the scene, so that the corresponding motion vector matches the actual movement undergone by that object from one frame to another, as illustrated in Figure 2.15. Evidently, this hypothesis breaks down for rotations, deformations, or even translational 3D movements. Besides, the boundaries of moving objects rarely conform to the rectangular grid imposed by BMC and recently uncovered areas might not have meaningful correspondences in previous frames. Nevertheless, BMC is known to work well even when its motion model is not accurate and, in spite of its shortcomings, BMC is still the most popular inter prediction technique to date. It is at the core of all current video coding standards.

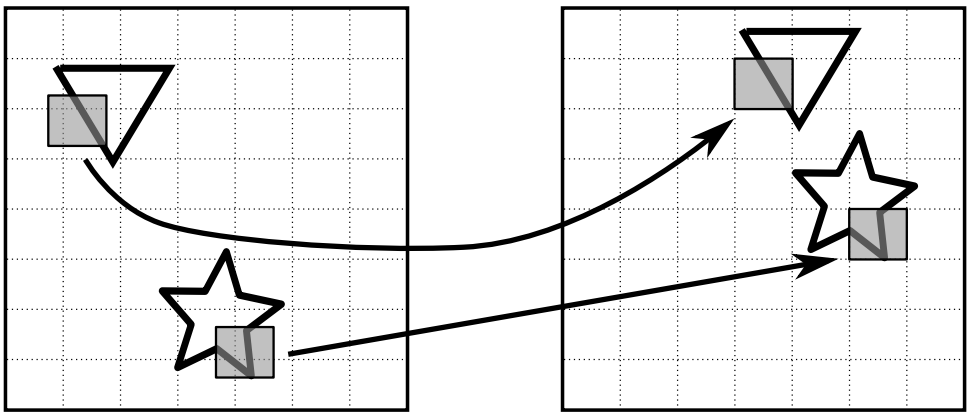

Figure 2.15: Translational motion hypothesis.

Both the matching criterion and the ME search algorithms are critical to the BMC approach rate-distortion (R-D) performance. The matching criterion defines in *what sense* an optimal prediction P_o matches its target T while the search algorithm defines which candidate blocks P are even tested for optimality. They also both have a great impact on the overall computational cost of an encoder, since ME is carried out for each target block in each frame.

2.4.1 Search Algorithms

For a fixed target T, the cost function $cost(P,T)$ is a function of the candidate block P only. Given such a cost function, whose minimization defines the matching criterion, an *ME algorithm* or *search algorithm* consist in a systematic way to find the prediction P_o which yields the minimum cost among the considered candidates.

We start by delimiting which candidates are considered for each target block. Excluding border considerations, this is usually done by defining a *search area* around the equivalent position of the target block in the previously coded frame. The center of the search area, located at the equivalent position r of the upper-left pixel of the target block T in the previously coded frame, defines the position of zero displacement, or zero motion vector. The search area itself is defined by displacements of up to $\pm w_x$ and $\pm w_y$ from r. That is, considered candidate blocks are all those blocks P whose upper-left pixel is at $r + \nu$, each respective motion vector $\nu = (\nu_x, \nu_y)$ satisfying $-w_x \leq \nu_x \leq w_x$ and $-w_y \leq \nu_y \leq w_y$.

Given a search area and a cost function, the most straightforward ME algorithm is the *full search algorithm* (FS) [4, 7]. It simply visits every single candidate block, calculating the costs for each of them while keeping track of the minimum value and its respective motion vector. The FS algorithm is guaranteed to find a global minimum for the cost within a fixed search area, irrespective of the visiting order, though the actual motion vector might change with the visiting order if the minimum is not unique. This algorithm, however, can be too computationally expensive for some applications since the cost function must be evaluated $(2w_x + 1)(2w_y + 1)$ times for *each* target block in each frame of the sequence.

Several ME algorithms, provide different level of trade-off between rate-distortion (R-D) performance and computational cost, with many providing R-D performance very close to the FS algorithm at a fraction of its time. In fact, given a fixed time budget for ME, some of these algorithms might in fact surpass the FS algorithm on the long run by allowing greater search areas.

Examples of fast search algorithms include the *two-dimensional logarithmic search* [27], the *one-at-a-time search* [28], and the *three step search* [29]. All of these algorithms are based on the *quadrant monotonic model*, which assumes that the cost function is monotonically non-decreasing in every direction when moving away from the optimal point [7]. Each of them employs a different strategy to exploit this model and track for a local minimum.

2.4.2 Matching Criterion

In spite of its name, the goal of BMC is *not* to closely match the movement of objects in the scene, but actually to effectively predict a frame in a clearly defined sense, which is to allow for R-D efficient coding of the target block. With that in mind, we can devise an optimal matching criterion. That would be the minimization of the Lagrangean R-D cost for the residual. This approach, however, is too computationally expensive to be used in practice since it would require the actual coding and decoding of each candidate residual for every target block to find their true rates and distortions. A heuristic approach is usually taken instead.

If the energy left in the residual is the smallest possible, we can reasonably expect that most of the energy in the signal is accounted for by the prediction itself, at least as much as it can be. We can also reasonably expect that an already small residual would likely minimize the overall impact of the subsequent quantization process, as well as require relatively few bits to code what remains. This argument points to the minimization of the *mean square error* (MSE) as a reasonable matching criterion:

$$MSE(P,T) = \frac{1}{N}\sum_{i=1}^{N}(P(i) - T(i))^2 = \frac{1}{N}\sum_{i=1}^{N}E(i)^2, \tag{2.3}$$

in which $N = n \times m$ is the number of pixels in each block, n and m being their width and height, respectively, and $P(i)$ and $T(i)$ are the i-th pixel in the prediction and target blocks, respectively. For the minimization process, we can drop the $1/N$ factor and work with the *sum of squared differences* (SSD):

$$SSD(P,T) = \sum_{i=1}^{N}(P(i) - T(i))^2 = \sum_{i=1}^{N}E(i)^2. \tag{2.4}$$

The need for squaring operations in equation (2.4) might still make it too expensive for some applications. It is common to select instead with the *sum of absolute differences* (SAD) as a distortion measure:

$$SAD(P,T) = \sum_{i=1}^{N} |P(i) - T(i)| = \sum_{i=1}^{N} |E(i)|. \tag{2.5}$$

Both the SSD and the SAD in equations (2.4) and (2.5) yield the prediction that is the *closest* to target in some sense. Minimization of the SSD is equivalent to the minimization of the L^2 or *euclidian* distance between the prediction and target blocks, while the minimization of the SAD is equivalent to the minimization of the L^1 or *Manhattan* distance between them. They are the most popular distortion measures for motion estimation.

While the rate of the residual cannot be calculated for each candidate without actually coding it, the rate for each motion vector can be easily estimated or even exactly calculated depending on the entropy coding method used. Usually, it is weighted against the selected distortion measure to form the actual cost in the spirit of equation (2.2):

$$cost(P,T) = dist(P,T) + \lambda_{ME} R(\nu), \tag{2.6}$$

in which $dist(P,T)$ is either the SAD or the SSD for P and T, λ_{ME} is a weighting factor and $R(\nu)$ is the number of bits required to code the motion vector ν, the displacement between the positions of P and T in their respective frames [6].

2.4.3 Enhanced Inter-prediction and the Shifting Transformation

Several techniques were proposed over the years to improve on the basic BMC approach described in Section 2.4. These include alternative algorithms for ME [30, 31] or alternative matching criteria [32, 33] to either speed up ME or to boost its R-D performance, as well as techniques to expand [17, 16] or complement [34, 35] this basic approach itself. We have already briefly mentioned some of these improvements in Sections 2.2.2 and 2.3. We now introduce the one proposed approach to BMC that motivated our own work.

In 2012, Blasi *et al* proposed their *enhanced inter-prediction* (EIP) technique [9]. This technique aims at improving the R-D performance of the BMC approach by considering a set of *transformed* candidate blocks P' instead of the original candidate blocks P in the search area. In fact, each original candidate block P in the search area gives rise to an entire set of transformed candidate blocks P_x, formally given by

$$P_x = \Theta(P|x^1, x^2, \dots x^n) = \Theta(P|\mathbf{x}), \tag{2.7}$$

in which $\Theta(\cdot|x^1, x^2, \dots x^n)$ is an *invertible* parametric transformation with associated parameters $\mathbf{x} = (x^1, x^2, \dots x^n)$. Motion estimation and compensation carries on as usual with its selected matching criterion and ME algorithm, but for *each* candidate block P, we consider instead

$$P' = \Theta(P|\mathbf{x}_o), \tag{2.8}$$

in which $\mathbf{x}_o$ is the parameter set that minimizes the candidate cost. That is, for each candidate block, $\mathbf{x}$ is optimized and set to $\mathbf{x}_o$ so that $P' = \Theta(P|\mathbf{x}_o)$ satisfies $cost(P', T) \leq cost(\Theta(P|\mathbf{x}), T)$ for every valid parameter vector $\mathbf{x}$.

Note that if Θ becomes the identity transformation for some given $\mathbf{x}$, than we always have $cost(P', T) \leq cost(P, T)$, which might give rise to a residual $E' = P' - T$ that reduces distortion and requires fewer bits to code. However, once P'_o is found, its respective optimal parameter set $\mathbf{x}_o$ must also be coded into the bitstream along with the corresponding E'_o and ν'_o, so that the decoder can invert the transformation to recreate to appropriate prediction. It becomes readily apparent that the EIP technique is only effective if, on average, the extra bits needed to code $\mathbf{x}_o$ are offset by a residual that either actually requires sufficiently fewer bits to code, or sufficiently reduces distortion, or both. Furthermore, since an optimal $\mathbf{x}_o$ is calculated for *every* candidate block in the search area, the transformation $\Theta(P|\mathbf{x})$ must also be so that the optimization of $cost(\Theta(P|\mathbf{x}), T)$ in $\mathbf{x}$ for given P and T can be done efficiently, so as to keep the overall computational burden feasible.

A particularly effective transformation for the implementation the EIP approach is the shifting transformation (ST) [9], also proposed by Blasi *et al* along with the EIP itself. The ST is a single parameter transformation $\Theta(\cdot|s)$. The parametric candidate P_s is simply given by

$$P_s = \Theta(P|s) = P + s, \tag{2.9}$$

in which the sum is understood in the sense that the scalar parameter s is uniformly added to each element of P. The effectiveness of EIP with ST stems from the fact that there is a simple algorithm to find the optimal parameter s_o for each transformed candidate block $P' = \Theta(P|s_o)$, and from the fact that s_o can be effectively coded.

The actual ME algorithm devised by Blasi *et al*, also provided in their original work as a proof of concept for the EIP [9], did not completely substitute conventional BMC, but actually complemented it. For each ME operation, their algorithm keeps track of the optimal usual prediction P_o along with optimal shifted prediction P'_o, which amount to testing each candidate twice, each turn with a different approach. At the end of each ME operation, both optimal solutions are tested against each other in the sense of a cost function analogous to equation (2.6). The rate for s_o weighted into the cost P'_o as in:

$$cost(P'_o, T) = dist(P'_o, T) + \lambda_{ME} R(\nu') + \lambda_{shift} R(s_o), \tag{2.10}$$

in which $R(s_o)$ is the rate for an s_o shift and λ_{shift} is a suitably defined lagrangian parameter. The rate for a zero shift is also similarly weighted into the cost of P_o. The encoder then outputs the prediction of minimal cost between the two, which is analogous to "turning off" the EIP when it does not provide sufficient gains over conventional BMC. With this algorithm, it has been shown that EIP with ST can be integrated into the H.264/AVC framework to significantly enhance its performance [9]. However, compliance to the standard is lost due to the need to code the shifting parameter.

Chapter 3

Motion Compensation with Residue Dispersion Measures

In this chapter, we propose a new matching criterion for ME and develop a two-pass ME algorithm to exploit it alongside one of the usual matching criteria. Unlike the SSD and the SAD, our proposed matching criterion does not minimize the size of the residual in any sense. Instead, the dispersion of the residual is minimized. We begin by taking a closer look at the enhanced inter-prediction with the shifting transformation introduced earlier. As we now show, it already points towards the usefulness of the minimum dispersion prediction. Unlike the EIP, however, our approach does not require side information coded into the bitstream, making it immediately compliant to any coding standard based on the hybrid DPCM/DCT coding model with BMC.

3.1 Optimum Shift Parameter for EIP with ST

Blasi *et al* provided an efficient algorithm to calculate the optimum shift parameter s_o along with their EIP with ST proposal, given the SAD as a cost function for ME. We now proceed to show that this optimal solution for s_o can be given a somewhat closed form. Though this new solution is not any more efficient than the original in any practical sense, it does reveal a lot about the qualitative role of s_o in the effectiveness of the EIP with ST. We begin by retracing the derivation of their original algorithm up to the point where a key property of the optimal solution is disclosed. We then argue for a slightly different solution.

Consider first that the cost of a candidate P is given by the SAD with respect to T and that both P, T and $E = P - T$ consists of blocks of N pixels. The cost then is given by

$$cost(P,T) = \sum_{i=1}^{N} |P(i) - T(i)| = \sum_{i=1}^{N} |E(i)|, \tag{3.1}$$

in which $P(i)$, $T(i)$ and $E(i)$ refer to the i-th pixel in the P, T, and E blocks, respectively. Given that this cost function is invariant under any reordering of the values, the particular order in which

the single index i indexes the pixels in the two dimensional blocks is immaterial.

Since neither the SAD nor the form of $\Theta(\cdot|s)$ as per equation (2.9) depends on the ordering of the elements of P, T or E, we also assume, without loss of generality, that E is arranged in increasing order by the indexation in i, that is, $E(i) \leq E(j)$, $\forall i < j$. Consider now, given fixed P, T, and E, the cost for a shifted candidate block:

$$cost(s) = \sum_{i=1}^{N} |(P(i) + s) - T(i)| = \sum_{i=1}^{N} |E(i) + s|, \tag{3.2}$$

in which $cost(s)$, a function of the shifting parameter s alone, is shorthand notation for $cost(P_s, T)$ with fixed P and T, P_s given by equation (2.9). Note that $cost(P, T) = cost(0)$, so, the identity transformation is considered by the ST. We now evaluate $cost(1)$, the cost for a candidate block with *positive* unitary shift. Let N_- be the number of negative entries in the *original* residual, E. Similarly, let N_0 and N_+ be the numbers of zero and positive elements in E, respectively. Note that $N = N_- + N_0 + N_+$. The original candidate cost can then be written as

$$cost(0) = -\sum_{i=1}^{N_-} E(i) + \sum_{i=N_-+N_0+1}^{N} E(i), \tag{3.3}$$

while $cost(1)$ is then given by

$$cost(1) = -\sum_{i=1}^{N_-} (E(i) + 1) + \sum_{N_-+1}^{N_-+N_0} (1) + \sum_{i=N_-+N_0+1}^{N} (E(i) + 1). \tag{3.4}$$

After rearranging equation (3.4), we finally have

$$cost(1) = cost(0) - N_- + N_0 + N_+. \tag{3.5}$$

Comparing equations (3.5) and (3.3), we see that $cost(1) < cost(0)$ if, and only if

$$N_- > N_0 + N_+. \tag{3.6}$$

At this point, our derivation departs from the original work on the EIP with ST [9]. Adding N_- to both sides of inequality (3.6), we see that a positive unitary shift will reduce the cost for a candidate prediction block if, and only if

$$N_- > \frac{N}{2}. \tag{3.7}$$

Similarly, it can be shown that a negative unitary shift will reduce the cost for a candidate prediction block, i.e., $cost(-1) < cost(0)$ if, and only if

$$N_+ > \frac{N}{2}. \tag{3.8}$$

Note that both inequalities (3.7) and (3.8) are *strict* inequalities.

Suppose now that inequality (3.7) is true, which guarantees that inequality (3.8) is false. We then apply a positive unitary shift transform and are left with $P_1 = P + 1$ and $E_1 = E + 1$. We redefine N_-, N_0 and N_+ analogously to the way we did before, according to the new *shifted* residue E_1. Suppose then that condition (3.7) is still met. It means that applying a further unitary positive shift will further reduce the cost. Since

$$\Theta(\Theta(P, s_1), s_2) = \Theta(P, s_1 + s_2), \tag{3.9}$$

which can be trivially shown, it implies that

$$cost(2) < cost(1) < cost(0). \tag{3.10}$$

We can now iterate this process until condition (3.7) is no longer met, at which point we are left with the optimal shift parameter s_o, which is positive in this case, after exactly s_o iterations. We are guaranteed that, after the last step, condition (3.8) will also be left unsatisfied. Otherwise, condition (3.7) would not have been met before the last step in the first place. We could have proceeded in a similar fashion for a negative shift, had the condition (3.8) been true at the beginning, which would make condition (3.7) false.

The iterative algorithm just given will produce the optimal shift s_o. However, it involves $|s_o|$ recounts of N_-, N_0, and N_+, in addition to $|s_o| \times N$ unitary sums or subtractions for *every* candidate prediction block. It is basically a brute force search. Still, it provides us with a valuable piece of information.

Remember that we assumed the $E(i)$ to be sorted in ascending order. Since the uniform addition of a constant value does not disturb this property, the optimal residual values $E_{s_o}(i) = P_{s_o}(i) - T(i)$ is also sorted in ascending order. At the optimal shift value, *neither* conditions (3.7) or (3.8) will be satisfied by E_{s_o}. Supposing that N is odd, this implies that the middle element in $E_{s_o} = E + s_o$ will be *zero*. That is, if N is odd, s_o is unique and it is simply given by $s_o = -E(\frac{N+1}{2}) = -\tilde{E}$, the negative of the sample *median* of the entries in E. If N is even, the solution is no longer unique. If one is interested in the *smallest* $|s_o|$ to produce the optimal cost, which might be the case in EIP since the shift parameter must be coded separately, one should select $s_o = -E(\frac{N}{2}) < 0$ if $E(\frac{N}{2}) > 0$, $s_o = -E(\frac{N}{2} + 1) > 0$ if $E(\frac{N}{2} + 1) < 0$, or $s_o = 0$ otherwise. However, for N even, any value for s_o that makes *both* $-E_{s_o}(\frac{N}{2}) \leq 0$ *and* $-E_{s_o}(\frac{N+1}{2}) \geq 0$ will leave *both* conditions (3.7) *and* (3.8) unmet, yielding the optimal cost. That is, any s_o satisfying $-E(\frac{N}{2} + 1) \leq s_o \leq -E(\frac{N}{2})$ will result in the same optimal cost. In particular, the negative of the median,

$$s_o = -\tilde{E}, \tag{3.11}$$

which is simply the midpoint of this interval for even N, is still an optimal solution. That is, equation (3.11), in which $\tilde{E}$ is the suitably defined median, is valid for any N.

Equation (3.11) provides the optimal shift parameter when the cost function is given by the SAD. Consider now the case in which the cost is given by the SSD. The cost for a shifted candidate is analogously given by

$$cost(s) = \sum_{i=1}^{N}((P(i)+s) - T(i))^2 = \sum_{i=1}^{N}(E(i)+s)^2. \tag{3.12}$$

Taking the derivative of equation (3.12) with respect to s and setting the result to zero at the optimal shift s_o, we have

$$s_o = -\frac{\sum_{i=1}^{N} E(i)}{N} = -\bar{E}, \tag{3.13}$$

so that the optimal shift when the cost function is given by the SSD is simply the negative of the mean values of the residue.

3.2 Heuristics for Motion Compensation with Dispersion Measures

Equation (3.13) for the optimal shift parameter in the SSD case was already given in the original EIP paper in a slightly different form [9]. Equation (3.11), however, is somewhat more difficult to grasp from their original derivation, which is why we chose to retrace it in the previous section. We believe it reveals a lot on *why* EIP with ST is effective.

Consider the EIP with ST while using the SAD as the distortion measure. By equation (3.11), each candidate prediction P is transformed to $P' = P - \tilde{E}$, which implies that each respective residual E is transformed to $E' = E - \tilde{E}$. Substituting E' into equation (2.5), we have

$$SAD_{shift} = \sum_{i=1}^{N} |E(i) - \tilde{E}|. \tag{3.14}$$

The shifted SAD is still a function of the residual alone, but it no longer measures the size of the residual in any sense. It is now proportional to the mean absolute deviation of the residual from its median. Analogously, using the SSD as the distortion measure, we get

$$SSD_{shift} = \sum_{i=1}^{N} \left(E(i) - \bar{E}\right)^2, \tag{3.15}$$

which is also still a function of the residual alone but, again, also not a measure of its size. The shifted SSD is proportional to the mean squared deviation of the residual from its mean. Both the median and the mean are measures of the *central tendency* of the residual, i.e., they are estimates of a central value around which the sample values in the residual tend to cluster. That makes both

equations (3.14) and (3.15) measures of *dispersion* of the residual, i.e., they both measure how much the sample values of the residual are spread around its central tendency. In fact, equation (3.15) is clearly proportional to the usual sample variance.

We can now see that, instead of testing a larger set of candidate blocks in search of the prediction that is the closest to the target in the sense of either the SAD or the SSD, the EIP with ST effectively tests the *same* set of candidate blocks in search for the minimum dispersion residual. In other words, the EIP with ST only changes the matching criterion for ME, so that the prediction that results in the most concentrated residual is chosen instead of the one that generates the smallest residual. Note that the residual of minimum dispersion can in fact be quite large in terms of the SAD or the SSD, if its median or mean values are large, respectively.

For an intuitive understanding of why the prediction of minimal residue dispersion can be more efficient than the well established prediction of minimal residue size, consider the artificial example in Figure 3.1. Given the target block T, candidate blocks P_1 and P_2 generate the candidate residual blocks E_1 and E_2, respectively. The minimization of either the SAD or the SSD will lead to the choice of P_1 as the prediction for T, since it clearly results in the smallest residual. However, candidate P_2 leads to a completely flat residual, which can be entirely coded in the single DC coefficient of the residue DCT. Furthermore, although there are no AC coefficients in the residual block E_2 itself, there are many AC details in the target block T. These details will be entirely preserved by the prediction P_2 alone, since none of it will be lost in the quantization of E_2.

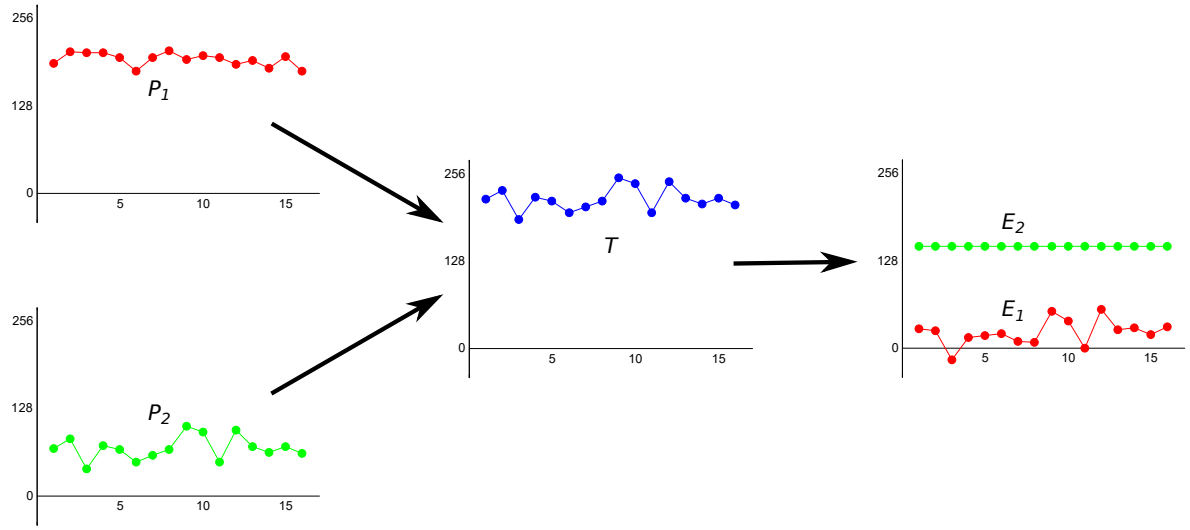

Figure 3.1: The advantages of dispersion minimization. Candidate P_1 is the prediction of minimal residue size, while P_2 is the prediction of minimal residue dispersion. Clearly, in this contrived example, residual E_2 can be coded more efficiently than E_1.

The extreme situation in the contrived example of Figure 3.1 is probably not representative of a typical coding scenario. However, it does reveal an important advantage of the prediction of minimal residue dispersion which is true in general. Since most modern codecs follow the DPCM/DCT coding model, the residue block is usually first transformed by some DCT-like transform and only then quantized before being actually encoded into the bit-stream. When we use either equations (3.14) or (3.15) to choose a prediction block, the DC coefficient looses its relative importance on

that choice, so the AC coefficients of the target block are better matched. It might imply less nonzero coefficients to be coded and possibly a smaller loss of texture details. Furthermore, a smaller sample dispersion indicates a smaller sample entropy, which might also imply a smaller number of bits needed to encode the block.

3.3 Compliant H.264/AVC Implementation

Equations (3.14) and (3.15) show that the effectiveness of the EIP with ST is already a compelling reason to consider predictions of minimal residue dispersion. To further consolidate their usefulness within the hybrid DPCM/DCT with BMC approach to video compression, we now devise a simple technique to integrate predictions of minimal residue dispersion into the H.264/AVC framework. Our proposed technique is fully compliant to the standard, so that no decoder adaptations are needed.

3.3.1 Proposed Dispersion Measure: The TADM

Though equation (3.14) is shown to be optimal in some sense, it is computationally expensive to evaluate the median $\tilde{E}$. Even the most efficient algorithms require at least a partial sorting of the residue values. As the cost function, we propose the *total absolute deviation from the mean* (TADM), given by

$$TADM = \sum_{i=1}^{N} |E(i) - \bar{E}|, \tag{3.16}$$

in which N is the number of pixels in a block. The TADM measures the absolute deviation of the residue from its central tendency like equation (3.14), but it uses the mean value $\bar{E}$ as a measure of its central tendency instead of the median. We chose this measure for its simplicity. Note that it requires neither a sorting of the residue values like equation (3.14) nor a squaring of every term like equation (3.15).

3.3.2 Practical Considerations

Though sub-optimal in the EIP sense, the mean $\bar{E}$ is almost as efficient as the median $\tilde{E}$ when used as the shift parameter in the EIP framework, as Table 3.1 shows. Performance is given in terms of the *BD-rate* [36], a measure of the average percent differences in rate between two rate-distortion curves for a given PSNR interval, indicating which offers a better trade-off between rate and distortion. More details on the BD-rate measure in Chapter 4. Negative BD-rate values indicate more efficient codding on the average. Column EIP in Table 3.1 shows the BD-rate savings for the original EIP with its original algorithm, while column EIP-DC shows the BD-rate savings for the EIP with shift parameter given by the mean value of the residue block, which amounts to using the TADM as given in equation (3.16) as the distortion function for ME. Both the EIP and

Table 3.1: BD-rates for EIP and EIP-DC, both against the conventional H.264 codec. Time savings are for the EIP-DC algorithm with respect to the EIP algorithm.

Sequence			**BD-Rate(%)**		**Time**
Name	**Resolution**	**FPS**	**EIP**	**EIP-DC**	**Savings**
mother-daughter	352x288	30	-6.13	-5.40	27%
crew	352x288	30	-9.43	-9.10	25%
mobile	352x288	30	-0.13	-0.07	19%
foreman	352x288	30	-3.48	-3.26	23%
RaceHorses	832x480	30	-2.99	-2.75	24%
PartyScene	832x480	50	-1.60	-1.45	18%
Mean			-3.96	-3.67	23%

the EIP-DC were implemented in a modified JM Reference Software [10], and both their BD-rate performances were calculated with respect to the unmodified JM H.264 encoder. Configurations in all three cases were set to use the full search algorithm with a single reference frame and variable length source coding. The time savings in Table 3.1 refer to the mean difference between the encoding times of the EIP-DC end the EIP algorithms with respect to the mean encoding time of the EIP algorithm. Note that the savings in time for using the mean instead of the median are very significant, compensating for the slightly worse coding efficiency.

We noted earlier that the potential advantage of a prediction of minimal dispersion residue is that it might better match the AC coefficients of the target, thus improving coding performance in the hybrid DPCM/DCT coding model. However, the size of the transform blocks in the hybrid framework need not conform to the size of the motion compensation blocks. For instance, the size of a macroblock in the H.264/AVC standard, its basic motion compensation unit, is fixed to 16×16 pixels. For each macroblock, BMC can be carried out for sub-partitions of size 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 or 4×4, allowing for BMC of variable block size. The actual partition mode chosen for each macroblock is decided by the encoder, usually in an R-D sense. For residue encoding, however, a complete 16×16 residual macroblock is divided in fixed partitions of size 4×4 which are then DCT-transformed and quantized, regardless of the size of the blocks actually used in ME.

To better exploit the relationship between the BMC block size and the DCT coding block size, we further modify the TADM to measure dispersion within 4×4 sub-blocks of each candidate prediction block as in

$$TADM = \sum_{j=1}^{N_{sb}} \sum_{i=1}^{16} |E_j(i) - \bar{E_j}|, \tag{3.17}$$

in which $N_{sb} = \frac{N}{16}$ is the number of 4×4 sub-blocks in each candidate prediction block and $\bar{E_j}$ is the mean of the j-th 4×4 sub-block E_j within each candidate prediction block. The H.264/AVC standard might also allow an optional 8×8 DCT for residual quantizing and coding, depending on the profile used. In this case, the TADM is analogously computed within 8×8 sub-blocks for each

Table 3.2: BD-rate EIP-DC-PURE against the conventional H.264 codec.

Sequence			BD-Rate(%)
Name	**Resolution**	**FPS**	**EIP-DC-ONLY**
mother-daughter	352x288	30	-1.43
crew	352x288	30	-6.21
mobile	352x288	30	3.55
foreman	352x288	30	3.33
RaceHorses	832x480	30	0.69
PartyScene	832x480	50	1.02
Mean			0.16

candidate prediction block. Henceforth, when we speak of the TADM, we mean either formula (3.17) or its 8×8 variant.

Furthermore, it should be noted that a simple naive substitution of the SAD by the TADM in the H.264 codec is not enough to improve its coding efficiency. In fact, not even the EIP can consistently improve the coding efficiency, even given its non-standard source coding dedicated to the shift parameter. Both the EIP and the EIP-DC of Table 3.1 follow the complementary approach given at the end of Section 2.4.3, which is, in essence, a two-pass algorithm. Unlike them, the EIP-DC-PURE in Table 3.2 was modified to *always* select the shifted prediction, which amounts to implementing the TADM alone, without the complementary use of the SAD. Testing conditions were the same as those for Table 3.1. Comparing Tables 3.2 and 3.1, we see that the complementary use of the SAD is crucial for a consistent performance of the EIP with ST approach.

Though given only for a small sample, the almost insignificant mean gain of the EIP-DC-PURE in Table 3.1, actually mean loss, seems to suggest that minimal residue dispersion prediction is not in itself superior to the smallest residue prediction, neither it is clearly inferior. Bear in mind that nothing else in the codec was adjusted for the new BMC distortion measure. In particular, mode decision functions and parameters such as λ_{ME} in (2.6) are still fine-tuned for the original SAD distortion measure. Even then, when we look at the performance of EIP-DC-PURE in each individual sequence, we see a wide spread in the BD-rate performances, with large gains in some sequences and large losses in others. This spread suggests that the minimum TADM and the minimum SAD matching criteria give two significantly different accounts for the motion in the sequence. The algorithm we propose, then, consists in a simple technique to exploit both accounts.

3.3.3 Proposed Algorithm: The DMCA

As stated before, in light of equations (3.14) and (3.15), the EIP with ST is already a proof of concept in favour of minimal dispersion residue prediction. We now present an algorithm to integrate our proposed minimal dispersion BMC approach into the hybrid DPCM/DCT coding framework. In theory, this integration can be done seamlessly into any BMC-based hybrid codec, maintaining full compliance to any standard based in this hybrid model. Experimental results follow in Chapter 4 for a fully compliant H.264 implementation. It serves both as further proof

of concept in favour of minimal dispersion residue prediction in general and as a case study for a TADM-based cost function.

We propose a two-pass algorithm, henceforth referred to as DMCA, standing for '*double matching criterion algorithm*. It produces two predictions for each macroblock, one with the original SAD distortion function and another one with the TADM instead. The SSD can be used instead of the SAD, with a total variance suitably defined in analogy to (3.17) instead of the TADM. The encoder then outputs the better one in a true R-D sense. No other functionality need to be modified in the TADM pass, neither does any encoder parameter value, though it is reasonable to to believe that doing so might improve the performance of the DMCA.

The algorithm is summarized in the pseudo-code below. Note that the macroblock predictions M_{SAD} and M_{TADM} are completely independent, not only in their motion estimation but also in their mode decision. That is, M_{SAD} and M_{TADM} can differ not only in their motion vectors but also in their partition modes. The rate-distortion cost J for each macroblock prediction M is evaluated by

$$J(M) = D(M) + \lambda R(M), \tag{3.18}$$

in which $D(M)$ and $R(M)$ are the overall distortion and the overall rate implied the prediction M, respectively, and λ is a Lagrange multiplier. Note that the distortion $D(M)$ is the *real* distortion introduced by the actual quantization of the residual, and the rate $R(M)$ is the actual rate needed to code M, including mode signalling, motion vector encoding for each sub-block, and quantized residual encoding. The DMCA then encodes only the best macroblock prediction in terms of the cost J into the bit-stream.

Algorithm: DMCA
FOR each macroblock
$M_{SAD} \leftarrow$ Macroblock prediction using the SAD distortion measure only
$M_{TADM} \leftarrow$ Macroblock prediction using the TADM distortion measure only
IF $J(M_{TADM}) < J(M_{SAD})$
Write M_{TADM} to the bit-stream
ELSE
Write M_{SAD} to the bit-stream

Observe that the decoder cannot know and need not know whichever of the macroblock predictions M_{SAD} or M_{TADM} is chosen by the encoder. Both contain all the information needed for decoding. Only the ME decision function is changed in each pass, but the compensation step and the encoding process for each macroblock prediction is rigorously the same. There is no need for the encoding of additional parameters nor of any side information at all. Therefore, the implementation of the DMCA at the encoder side requires no modifications at the decoder side, thus making it compliant to the H.264/AVC coding standard.

Notice the similarities and differences between the DMCA and the complementary SAD/EIP algorithm of Section 2.4.3. They both test a smallest residue prediction and minimal residue dispersion prediction. However, the latter tests these predictions against each other at every ME operation, which might result in macroblock with both types of prediction. The DMCA, on the other hand, produces two different predictions for the entire macroblock, including mode decision, with a different yet fixed matching criterion for ME in each pass. Also, since minimal SAD and the minimal DATM solutions are tested against each other only once, the true R-D cost as in (2.2) can be used instead of the estimated cost (2.10). Finally, unlike the SAD/EIP algorithm, only the cost function for candidate selection is changed, not the resulting residue itself. No additional parameters must be coded into the bitstream then, making possible a compliant implementation into any BMC hybrid coding standard.

Chapter 4

Experimental Results

In this chapter, we test the performance of the double matching criterion algorithm proposed in Section 3.3.3 against the reference H.264/AVC codec. The algorithm is tested for a large number of sequences with varying characteristics for consistent gains in coding performance in a variety of testing conditions.

4.1 Experimental Settings

For testing purposes, the DMCA was integrated into the JM reference software [10] for the H.264/AVC standard. Only the encoder had to be modified, since the resulting bit-stream is rigorously compliant to the standard.

Our implementation was tested on several popular test sequences in their full length. These sequences are identified in Table 4.1 and their corresponding tags will be used to reference them henceforth. To ensure that our tests cover a wide range spatial and temporal characteristics, each sequence was tested for their spatial perceptual information (SI) and for their temporal perceptual information (TI) [37]. The SI and TI measures try to encode the amount of the spatial and temporal activities of an entire sequence in a single number, respectively, by measuring the amount of variation in pixel values in space and time. The spacial and temporal perceptual content for each sequence tested is given in Figure 4.1 in terms of SI and TI, where we can see that the selected test sequences cover a broad range of characteristics.

In order to assess the performance of a video encoder for a given sequence, we need to evaluate its effectiveness in the trade-of between the quality of the reconstructed test sequence and its compressed bit-rate. One way to do that is to evaluate the PSNR between the reconstructed and original sequences for several distortion operation points and plot it against their respective bit-rates. We take that route in Figure 4.2, where we can see a sample result from the first test in the next section. The R-D curve is interpolated from four QP operation points for both the original JM encoder and the modified encoder with the DMCA. As expected, for each QP value, the DMCA has an operation point slightly above and to the right of the respective operation point for original JM, resulting in an R-D curve that offers a better trade-off between rate and distortion. However,

Table 4.1: Sequences used throughout the tests in this chapter.

Resolution	**FPS**	**TAG**	**Name**
352x288	30	S01	city
		S02	crew
		S03	foreman
		S04	harbour
		S05	mobile
		S06	mother-daughter
		S07	soccer
832x480	30	S08	RaceHorses
		S09	Mobisode2
832x480	50	S10	BasketballDrill
		S11	PartyScene
704x576	60	S12	city
		S13	crew
		S14	harbour
		S15	soccer
1920x1080	24	S16	Kimono1
		S17	ParkScene
		S18	Tennis
1920x1080	50	S19	Cactus
		S20	Crowdrun
		S21	DucksTakeOff
		S22	ParkJoy
		S23	RushHour

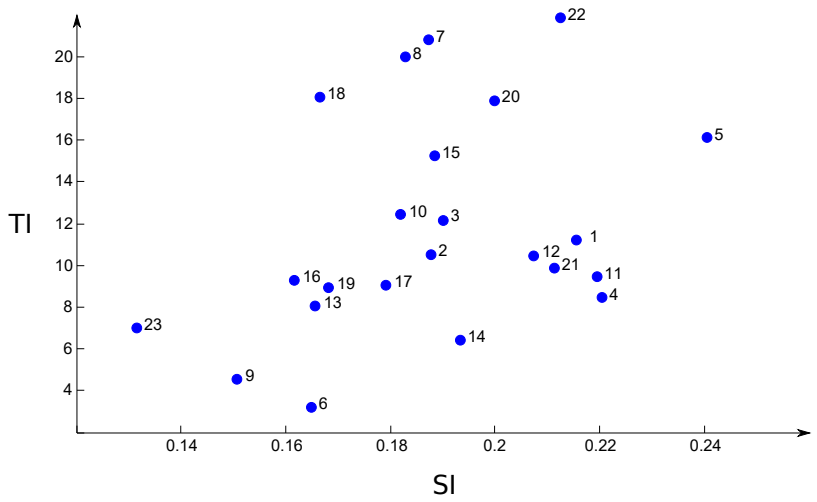

Figure 4.1: Motion content of tested sequences.

Figure 4.2 offers little insight into *how much* the DMCA is more efficient than conventional motion estimation. Besides, huge collections of such curves quickly become cumbersome and make it difficult to evaluate the overall performance gains for a large set of test figures.

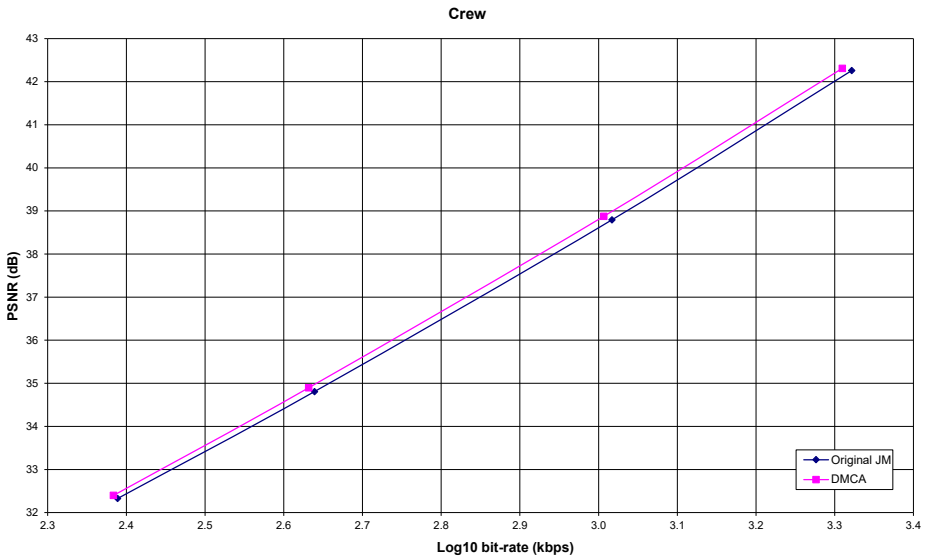

Figure 4.2: Typical test result. Curve for sequence S02 under the test conditions of the first experiment in Section 4.2.

In order to overcome these difficulties, results for the DMCA are given in terms of the BD-rate [36] against the unmodified JM encoder. For each sequence, the BD-rate measures the mean bit-rate difference in percent values between the test R-D curve and an anchor R-D curve over an interval of PSNR values, thus expressing the comparative improvement over the R-D trade-off in a single number. As in Figure 4.2, the operation points for both the test encoder and the anchor encoder are evaluated at four different QP values. Their respective R-D operation points are then interpolated for the calculation of the average difference in percent values over the full PSNR range covered by the interpolated curves. Negative values express performance gains, while positive values express performance loss. It is possible that the test and anchor R-D curves cross

each other in the tested region, meaning that one is better than the other for some rate range but worse in the remaining range. This behaviour cannot be captured in a single number. This setback can be somewhat mitigated by taking the average only over the low rates range of the curves, between the two operation points of higher QP values, and again only over the high rates range of the curves, between the two operation points of lower QP values. Together, the BD-rates for the full range, low rates range, and high rates range provide a good description of the comparative performance of two encoders for a given sequence. For each experiment, each encoder was tested on all of the sequences in Table 4.1 at four different QP values, namely, 22, 27, 33, and 37.

4.2 Results

For the first experiment, the encoder was set to use exclusively P frames after the first IDR frame, with five reference frames for ME and CABAC entropy coder. Intra modes were not allowed for P slices, but skip mode was considered. Results are shown in Table 4.2.

The proposed technique consistently outperforms the unmodified JM encoder in every sequence tested. Results show gains of up to 3,81% and at least 0,70% on these sequences, with an average 2,04% gain. Results also show that considering the TADM leads to consistent gains over every rate range, with higher gains being observed in the high-rates range for most tested sequences.

Table 4.3 compares the DMCA with absolute deviation from mean and with absolute deviation from the median, both for 4×4 sub-blocks as in equation (3.17). Unlike in the EIP framework, where Table 3.1 shows that the deviation from the median is generally more efficient than the deviation from the mean, the BD-rates of Table 4.3 show that the deviation from the mean is consistently more efficient than the deviation from the median in the DMCA framework.

The DMCA technique is similar to the optional *multiple QP testing* (MQPT), available in the original JM codec. In fact, much of the code for the MQPT was reused in our implementation of the DMCA. Much like the DMCA, the MQPT technique works by independently predicting each macroblock in multiple passes, then encoding only the one prediction that performs best in the R-D sense. As the name suggests, each pass of this technique tests a different QP value for motion estimation and mode decision. In Table 4.4 we compare the performances of the unmodified JM encoder with 2 and 3 QP values for MQPT, given in columns **MQPT-2** and **MQPT-3**, respectively, and the performance of the DMCA with a single QP tested, all three against the original JM with a single QP. Also displayed in Table 4.4 is the savings in coding time for the DMCA with respect to **MQPT-3**. These time saving refer to the mean difference in encoding time between the DMCA and the **MQPT-3** algorithms with respect to the mean encoding time for the **MQPT-3** algorithm. The MQPT method in the JM codec is only available when the *rate-distortion optimized quantization* (RDOQ) [38, 39] is activated, so all four tests were performed with RDOQ, unlike previous tests. Remaining configurations were held the same. We can see that the DMCA consistently outperforms the triple-pass MQPT-3 while spending significantly lower computation time.

Table 4.2: BD-rates of DMCA with TADM against conventional H.264/AVC.

TAG	**BD-Rate**		
	Full Range	**Low Rates**	**High Rates**
S01	-0.70	-0.81	-0.72
S02	-3.75	-3.92	-3.27
S03	-2.52	-2.62	-2.06
S04	-1.06	-0.98	-1.11
S05	-0.83	-0.65	-1.08
S06	-1.50	-2.00	-1.31
S07	-1.38	-1.32	-1.58
S08	-2.27	-1.97	-2.36
S09	-3.81	-4.88	-3.07
S10	-3.13	-3.33	-2.75
S11	-1.32	-1.10	-1.51
S12	-1.09	-1.35	-0.59
S13	-3.25	-3.12	-2.80
S14	-1.69	-1.62	-1.66
S15	-1.59	-1.82	-1.29
S16	-3.70	-4.31	-3.05
S17	-1.46	-1.56	-1.40
S18	-2.05	-2.84	-1.36
S19	-3.07	-3.43	-2.43
S20	-1.68	-1.51	-1.85
S21	-1.71	-1.05	-2.56
S22	-0.90	-0.80	-1.06
S23	-2.55	-3.88	-1.93
Mean	-2.04	-2.21	-1.86

Table 4.3: BD-rates for DMCA with absolute deviation from mean and with absolute deviation from the median, both against the conventional H.264 codec and both in the full range

TAG	**BD-Rate**	
	Mean	**Median**
S01	-0.70	-0.60
S02	-3.75	-3.75
S03	-2.52	-2.45
S04	-1.06	-0.98
S05	-0.83	-0.71
S06	-1.50	-1.51
S07	-1.38	-1.20
S08	-2.27	-2.19
S09	-3.81	-3.50
S10	-3.13	-3.08
S11	-1.32	-1.25
S12	-1.09	-1.02
S13	-3.25	-3.28
S14	-1.69	-1.70
S15	-1.59	-1.55
S16	-3.70	-3.84
S17	-1.46	-1.42
S18	-2.05	-2.01
S19	-3.07	-3.02
S20	-1.68	-1.63
S21	-1.71	-1.70
S22	-0.90	-0.84
S23	-2.55	-2.61
Mean	-2.04	-1.99

Table 4.4: BD-rates for JM with 2 and 3 QP values tested and for DMCA, all against the conventional H.264 codec with a single QP pass. Unlike previous tests, RDOQ was used in all four cases. All BD-rates given for the full range only. Time saving are for the mean encoding time of the DMCA against the mean encoding time for the MQPT-3

TAG	**BD-Rate (%)**			**Time**
	MQPT-2	**MQPT-3**	**DMCA**	**Savings**
S01	-0.16	-0.87	-0.99	14%
S02	0.16	-1.35	-3.56	13%
S03	-0.10	-1.23	-2.32	14%
S04	-0.01	-0.15	-1.01	15%
S05	-0.03	0.21	-0.84	15%
S06	-0.44	-1.41	-1.76	18%
S07	-0.15	-1.50	-1.46	12%
S08	-0.09	-0.92	-2.05	14%
S09	-0.07	-2.52	-4.17	21%
S10	0.06	-0.92	-2.92	16%
S11	0.02	-0.49	-1.20	17%
S12	0.03	-0.04	-1.03	15%
S13	0.02	-2.18	-3.16	17%
S14	-0.09	-0.77	-1.78	17%
S15	-0.05	-1.64	-1.62	14%
S16	0.14	-3.65	-3.50	27%
S17	0.02	-1.77	-1.34	26%
S18	0.02	-2.80	-1.97	23%
S19	-0.42	-1.85	-2.85	18%
S20	-0.09	-0.82	-1.76	13%
S21	-0.20	-0.54	-1.61	20%
S22	-0.01	-0.24	-0.94	15%
S23	0.08	-3.70	-2.56	15%
Mean	-0.06	-1.35	-2.02	17%

Table 4.5: BD-rates of DMCA with TADM against conventional H.264/AVC, now with weighted prediction and biprediction allowed in both cases as well as varying transform block size.

TAG	**BD-Rate (%)**		
	Full Range	**Low Rates**	**High Rates**
S01	-1.67	-1.55	-1.54
S02	-3.67	-3.75	-3.66
S03	-2.65	-2.39	-2.80
S04	-0.71	-0.58	-1.10
S05	-0.60	-0.73	-0.61
S06	-2.84	-2.71	-2.97
S07	-1.30	-1.21	-1.27
S08	-1.28	-1.35	-1.40
S09	-4.22	-5.19	-3.37
S10	-3.30	-4.04	-2.35
S11	-1.18	-1.31	-1.12
S12	-1.00	-1.19	-0.92
S13	-3.32	-2.99	-3.46
S14	-1.30	-1.02	-1.64
S15	-1.28	-1.30	-1.41
S16	-3.19	-3.04	-3.33
S17	-1.62	-1.62	-1.53
S18	-2.11	-2.87	-1.34
S19	-3.52	-3.82	-3.03
S20	-1.17	-1.08	-1.43
S21	-0.82	-0.82	-0.82
S22	-0.70	-0.80	-0.62
S23	-2.88	-3.06	-2.84
Mean	-2.01	-2.11	-1.94

Finally, we test the DMCA against the unmodified JM encoder in a more general setting. Results are shown in Table 4.5. For this final testing, weighted prediction and biprediction were allowed, as well as local decisions between the 4×4 and the 8×8 transform blocks. Up to 5 frames were used as reference for motion compensation and each P frame is followed by 7 B frames, with B frames allowed to be used as references. Equation (3.17) was suitably modified when 8×8 blocks were tested and applied to the final residual of each candidate prediction *after* weighted prediction and biprediction were applied. Results also show consistent gains for the DMCA in combination with these other techniques.

4.3 Analysis

Results in Table 4.2 shows that the DMCA does indeed improve the coding efficiency of the BMC approach, both consistently and significantly, for sequences in a wide range of characteristics as per Figure 4.1. Besides, Table 4.3 also shows that there is no loss in coding performance if the mean is used as a measure of central tendency for the residue instead of the median, which is optimal in the EIP sense. In fact, though the mean was primarily chosen for its lower computational cost, Table 4.3 actually shows that the mean is consistently superior to the median. That superiority indicates that the heuristic reasoning of Section 3.2 might actually be more effective than optimality in the EIP sense, thus providing further support for the minimal residue dispersion criterion for block matching .

With a suitable rate-distortion optimizing decision function, a two-pass algorithm like the DMCA can hardly *degrade* the coding performance, which might raise questions as to whether the performance gains are worthy the extra computational cost. Table 4.4 dismisses those questions, showing that the DMCA is both more reliable and more effective, as well as more cost-efficient than a similarly time-consuming multi-pass technique. Note that the intent of this limited experiment is to show the suitability of the DMCA, not to present it as a substitute for multiple QP testing. In fact, both techniques can be easily combined, possibly leading to further gains in performance.

Finally, Table 4.5 shows that the DMCA can still improve the BMC approach even when it is already enhanced by other techniques such as biprediction, weighted prediction, and RD-optimal transform size. In fact, comparing Tables 4.2 and 4.5, we perceive very similar performances, both in the mean and consistently throughout the test sequences. This preservation of performance gains in different experimental settings, also observed in Table 4.4 with the use of RDOQ, indicates that the DMCA can be effectively combined with a multitude of techniques. That is, the DMCA does not "compete" against other techniques for gains, indicating that its gains are from a different nature. The higher effectiveness of the DMCA in Table 4.4 when compared with multiple QP testing, which is a similar technique, seems to suggest that gains of the DMCA derive from a higher diversity of options for rate-distortion optimization. This better diversity of options seems to be observed even against the higher number of options in triple QP testing, thus reinforcing the heuristic reasoning of Section 3.3.3 which led to the development of the DMCA in the first place.

Chapter 5

Conclusions

The ever growing demand for video data presses for ever increasing video coding efficiency. The key to this efficiency requirement lies in the high temporal redundancy characteristic of most video signals. Block-based motion compensation has become the technique of choice for exploiting this redundancy.

Another key factor for the ubiquity of video services is standardization. Industry standards have allowed for the intercommunication of a wide rage of devices with different resources. Compliance to popular standards can decisively dictate the costs for the implementation of new techniques into already deployed equipment. In tune with the widespread adoption of the BMC technique, most modern video coding standards makes provision for its effective implementation.

In this work, we argue for the benefits of BMC informed by the dispersion of the residue values. As noted, the EIP with ST already lends a strong testimony to these benefits, albeit at the cost of coding specialized side information, which prevents its compliance to established coding standards. To further consolidate the importance of minimal residue dispersion as a matching criterion for ME, we present the DMCA, a two-pass technique to integrate the proposed TADM dispersion measure into the BMC framework without the need for specialized side information.

The DMCA is implemented in the JM reference software for the popular H.264/AVC coding standard, for testing against the unmodified JM encoder. Full compliance to the H.264/AVC is maintained. Results show significant improvements over the original JM encoder with average 2.04% BD-rate gains, lending further support to our claim that ME can be improved by considering the dispersion of the residue. The TADM is primarily chosen for its relatively low computational cost, but it is also shown to outperform other dispersion measures in the DMCA framework.

Future research will include investigation for a single-pass algorithm, aiming for reducing computation time. Current results suggests that it is unlikely that the TADM will ever consistently outperform the SAD without joint consideration. However, a local low cost decision function for automatic switching of the matching criterion may be a viable solution. Moreover, more robust dispersion measures, as well as more sophisticated uses thereof, might bring about even higher gains, even though the proposed technique might be appealing on itself given its simplicity and its compliance to the H.264/AVC standard. Future work will also include research in that direction.

BIBLIOGRAPHIC REFERENCES

[1] BULL, D. R. *Communicating Pictures: A Course in Image and Video Coding.* [S.l.]: Academic Press, 2014.

[2] WALLACE, G. K. The JPEG still picture compression standard. *Communications of the ACM*, AcM, v. 34, n. 4, p. 30–44, 1991.

[3] SAYOOD, K. *Introduction to Data Compression.* [S.l.]: Morgan Kaufmann, 2012.

[4] CHAKRABARTI, I.; BATTA, K. N. S.; CHATTERJEE, S. K. *Motion Estimation for Video Coding.* [S.l.]: Springer, 2015.

[5] SULLIVAN, G.; WIEGAND, T. Video compression - from concepts to the H.264/AVC standard. *Proceedings of the IEEE*, v. 93, n. 1, p. 18–31, 2005.

[6] GUO, J. et al. A novel criterion for block matching motion estimation. In: *Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on.* [S.l.: s.n.], 1998. p. 841–844 vol.1.

[7] METKAR, S.; TALBAR, S. *Motion Estimation Techniques for Digital Video Coding.* [S.l.]: Springer, 2013.

[8] ARORA, S.; RAJPAL, N. Survey of fast block motion estimation algorithms. In: *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on.* [S.l.: s.n.], 2014. p. 2022–2026.

[9] BLASI, S.; PEIXOTO, E.; IZQUIERDO, E. Enhanced inter-prediction via shifting transformation in the H.264/AVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 23, n. 4, p. 735–740, 2013.

[10] Joint Model. H. 264/avc reference software. *http://iphome.hhi.de/suehring/tml/download.*

[11] RICHARDSON, I. E. *The H.264 Advanced Video Compression Standard.* [S.l.]: John Wiley & Sons, 2011.

[12] SALOMON, D. *Data Compression: The Complete Reference.* [S.l.]: Springer, 2007.

[13] COVER, T. M.; THOMAS, J. A. *Elements of Information Theory.* [S.l.]: John Wiley & Sons, 2012.

[14] TAUBMAN, D. S.; MARCELLIN, M. W. JPEG2000: Standard for interactive imaging. *Proceedings of the IEEE*, IEEE, v. 90, n. 8, p. 1336–1357, 2002.

[15] GIORDA, F.; RACCIU, A. Bandwidth reduction of video signals via shift vector transmission. *Communications, IEEE Transactions on*, IEEE, v. 23, n. 9, p. 1002–1004, 1975.

[16] BROFFERIO, S.; ROCCA, F. Interframe redundancy reduction of video signals generated by translating objects. *Communications, IEEE Transactions on*, IEEE, v. 25, n. 4, p. 448–455, 1977.

[17] WIEGAND, T.; ZHANG, X.; GIROD, B. Long-term memory motion-compensated prediction. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 9, n. 1, p. 70–84, 1999.

[18] WIEGAND, T. et al. Rate-constrained coder control and comparison of video coding standards. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 13, n. 7, p. 688–703, 2003.

[19] ORTEGA, A.; RAMCHANDRAN, K. Rate-distortion methods for image and video compression. *Signal Processing Magazine, IEEE*, v. 15, n. 6, p. 23–50, 1998.

[20] SULLIVAN, G.; WIEGAND, T. Rate-distortion optimization for video compression. *Signal Processing Magazine, IEEE*, v. 15, n. 6, p. 74–90, 1998.

[21] NOCEDAL, J.; WRIGHT, S. *Numerical Optimization*. [S.l.]: Springer Science & Business Media, 2006.

[22] WIEGAND, T.; GIROD, B. Lagrange multiplier selection in hybrid video coder control. In: *Image Processing, 2001. Proceedings. 2001 International Conference on*. [S.l.: s.n.], 2001. p. 542–545 vol.3.

[23] READER, C. History of MPEG video compression-ver. 4.0. *Joint Video Team (JVT), JVT-E066*, 2002.

[24] WIEGAND, T. et al. Overview of the H.264/AVC video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, v. 13, n. 7, p. 560–576, 2003.

[25] SULLIVAN, G. J.; TOPIWALA, P. N.; LUTHRA, A. The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Optical Science and Technology, the SPIE 49th Annual Meeting*. [S.l.], 2004. p. 454–474.

[26] ITU-T RECOMMENDATION. H.264 advanced video coding for generic audiovisual services. *ISO/IEC*, 2014.

[27] JAIN, J. R.; JAIN, A. K. Displacement measurement and its application in interframe image coding. *Communications, IEEE Transactions on*, IEEE, v. 29, n. 12, p. 1799–1808, 1981.

[28] CHEN, M.-J.; CHEN, L.-G.; CHIUEH, T.-D. One-dimensional full search motion estimation algorithm for video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 4, n. 5, p. 504–509, 1994.

[29] SRINIVASAN, R.; RAO, K. Predictive coding based on efficient motion estimation. *Communications, IEEE Transactions on*, IEEE, v. 33, n. 8, p. 888–896, 1985.

[30] HSIEH, C.-H. et al. Motion estimation algorithm using interblock correlation. *Electronics Letters*, IET, v. 26, n. 5, p. 276–277, 1990.

[31] CHALIDABHONGSE, J.; KUO, C. J. Fast motion vector estimation using multiresolution-spatio-temporal correlations. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 7, n. 3, p. 477–488, 1997.

[32] WANG, Y.; WANG, Y.; KURODA, H. A globally adaptive pixel-decimation algorithm for block-motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 10, n. 6, p. 1006–1011, 2000.

[33] JING, X.; ZHU, C.; CHAU, L.-P. Smooth constrained block matching criterion for motion estimation. In: *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*. [S.l.: s.n.], 2003. v. 3, p. III–661–4 vol.3.

[34] SULLIVAN, G. J. Multi-hypothesis motion compensation for low bit-rate video coding. In: IEEE. *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*. [S.l.], 1993. v. 5, p. 437–440.

[35] ORCHARD, M. T.; SULLIVAN, G. J. Overlapped block motion compensation: An estimation-theoretic approach. *Image Processing, IEEE Transactions on*, IEEE, v. 3, n. 5, p. 693–699, 1994.

[36] BJONTEGAARD, G. Improvements of the BD-PSNR model. *ITU-T SG16 Q*, v. 6, p. 35, 2008.

[37] OSTASZEWSKA, A.; KLODA, R. Quantifying the amount of spatial and temporal information in video test sequences. In: *Recent Advances in Mechatronics*. [S.l.]: Springer, 2007. p. 11–15.

[38] KARCZEWICZ, M. et al. RD based quantization in H.264. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *SPIE Optical Engineering+ Applications*. [S.l.], 2009. p. 744314–744314.

[39] WEN, J. et al. Fast rate distortion optimized quantization for H.264/AVC. In: *Data Compression Conference (DCC), 2010*. [S.l.: s.n.], 2010. p. 557–557.