UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

FELIPE MARTIN SAMPAIO

Energy-Efficient Memory Hierarchy for Motion and Disparity Estimation in Multiview Video Coding

Thesis presented in partial fulfillment of the requirements for the degree of Master of Computer Science

Prof. Dr. Sergio Bampi
Advisor

Prof. Dr. Luciano Volcan Agostini
Co-Advisor

Porto Alegre, Fevereiro de 2013.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Sampaio, Felipe

Energy-Efficient Memory Hierarchy for Motion and Disparity Estimation in Multiview Video Coding / Felipe Sampaio. – Porto Alegre: PPGC da UFRGS 2012.

120 f.:il.

Orientador: Sergio Bampi; Co-orientador: Luciano Agostini.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR - RS, 2013.

1. Codificação de Vídeo Multivistas. 2. Hierarquia de Memória 3. Consumo de Energia. I. Bampi, Sergio. II. Agostini, Luciano. III. Energy-Efficient Memory Hierarchy for Motion and Disparity Estimation in Multiview Video Coding.

AGRADECIMENTOS

Agradeço, inicialmente, aos meus pais e à minha irmã pelo apoio incondicional que me deram durante toda a minha jornada nestes dois anos de mestrado, desde a decisão pelo mestrado em outra cidade até a aceitação de nos vermos apenas aos finais de semana, quando possível. Sempre acreditaram na minha capacidade e no meu potencial. Agradeço tudo que sempre fizeram por mim! Amo vocês!

O mestrado, além do título de mestre e do meu ingresso imediato no doutorado, me proporcionou uma das maiores alegrias da minha vida: conhecer uma pessoa muito especial com quem eu estou dividindo minha vida, a minha namorada Meiri. Não existem palavras para agradecer o apoio, os conselhos e tudo que vivemos juntos (e o que vamos viver ainda). Quero te dizer que esta etapa da minha vida foi muito mais feliz porque tu, Meiri, estavas ao meu lado sempre. Te amo!

Minha vida na capital dos gaúchos é muito mais tranquila e divertida graças aos meus grandes amigos e colegas de apartamento Daniel e Diego. Com eles dividi minhas experiências, aprendi o espírito do coleguismo e cresci como pessoa e como ser humano. Obrigado pelo companheirismo! Além deles, o amigo e jornalista Mateus Kerr também merece um agradecimento especial, pelas noitadas de filosofia e de cervejas durante todo esse período. Tu és o cara, bruxo!

Mesmo morando agora em outra cidade, é impossível me esquecer daqueles com quem eu sempre posso contar para tomar aquele mate, jogar aquele futebol, ou simplesmente se ‘ajuntar’ numa noite de chuva para eu ganhar (com facilidade, sempre) no vídeo game. Agradeço a todos os meus amigos que estão espalhados agora pelo Brasil, mas que nunca se esquecem de te ligar pra te convidar para um churrasco.

Na UFRGS, encontrei um ambiente de trabalho ótimo, engraçado e de muita parceria. Com o tempo, mais que colegas de trabalho, eles tornaram-se amigos que serão para a vida inteira, não tenho dúvidas. Agradeço a todos os colegas que passaram pelo laboratório 215, todos foram importantes de alguma forma. De maneira especial, agradeço aos colegas: Duda, Cláudio, Daniel, Leonardo, Kléber e Cauane, pelas discussões filosóficas, churrascos e por toda a parceria.

Mesmo longe da UFPel, ainda assim continuei trabalhando com os colegas e amigos que deixei por lá. Então, um agradecimento especial a todos, principalmente para o Mateus com quem tive possibilidade de continuar trabalhando e trocando ideias.

Todo o meu trabalho e tudo que produzi nestes dois anos de mestrado só foram possíveis graças ao apoio e orientação dos professores e amigos Sergio Bampi e Luciano Agostini. Muito obrigado pelos ensinamentos e pelo apoio ao meu trabalho.

Cabe aqui um agradecimento especial ao meu orientador informal, com quem tive o prazer de trabalhar desde 2010 e que espero continuar trabalhando por muito tempo ainda. Zatt, tu és um cara fora do comum, e sabes que tens participação fundamental desde meu trabalho final da graduação até os trabalhos mais recentes. Obrigado pela parceria, por todas as conversas e pelos trabalhos intensos nesses últimos anos.

A qualidade das publicações e dos trabalhos foi possível também graças aos pesquisadores do KIT que colaboraram de maneira importante e que deram credibilidade ao meu trabalho. Obrigado ao Shafique e ao Professor Jörg Henkel pelas colaborações e trabalhos em conjunto que tivemos nos últimos anos.

Agradeço também aos órgãos de fomento brasileiros, em especial ao CNPq e à CAPES, que financiaram minhas atividades e minhas participações em conferências, onde pude divulgar minhas contribuições para a comunidade científica.

Por último, e não menos importante, eu agradeço ao Sport Club Internacional pelas alegrias e inúmeras vitórias que eu pude presenciar, agora estando fisicamente no Beira Rio, podendo cantar e te apoiar. Obrigado Campeão de Tudo!

SUMMARY

LIST OF ABREVIATIONS AND ACRONYMS

3D	Three-Dimensional
AGU	Address Generation Unit
AVC	Advanced Video Coding
BD-BR	Bjontegaard Delta – Bitrate
BD-PSNR	Bjontegaard Delta – Peak-to-Signal Noise Ratio
CABAC	Context-Adaptive Based Arithmetic Coding
DC	Disparity Compensation
DDR	Double Data Rate
DE	Disparity Estimation
DI	Disparity Index
FBC	Frame Buffer Compression
FPGA	Field Programmable Logic Array
FS	Full Search
GDV	Global Disparity Vector
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
ISO/IEC	International Organization for Standardization/ International Electrotechnical Commission
ITU	International Telecommunication Unit
ITU-T	ITU Telecommunication Standardization Sector
JPEG	Joint Photographic Experts Group
JPEG-LS	Joint Photographic Experts Group - Lossless
JM	Joint Video Coding Model
JMVC	Joint Multiview Video Coding Model
JVT	Joint Video Team
LPDDR	Low-Power Double Data Rate
MB	Macroblock
MBDR	Macroblock-centered Data Reuse

MC	Motion Compensation
ME	Motion Estimation
MMSQ-EC	Min-Max Scalar Quantization / Error Compensation
MRF	Multiple Reference Frames
MSE	Mean Squared Error
MVC	Multiview Video Coding
PDF	Probability Density Function
PSNR	Peak-to-Signal Noise Ratio
QP	Quantization Parameter
RCDR	Reference-Centered Data Reuse
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RFCAVLC	Reference Frame Context
RGB	Red, Green and Blue
SAD	Sum of Absolut Differences
SATD	Sum of Absolute Transformed Differences
SI	Spatial Index
SRAM	Static Random Access Memory
SSD	Sum of Squared Differences
SW	Search Window
TI	Temporal Index
TH	Threshold
VBS-ME	Variable Block Size Motion Estimation
VLC	Variable Length Coding
YCbCr	Luminance, Blue Chrominance and Red Chrominance

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This Master Thesis proposes a memory hierarchy for the Motion and Disparity Estimation (ME/DE) centered on the encoding references, called Reference-Centered Data Reuse (RCDR), focusing on energy reduction in the Multiview Video Coding (MVC). In the MVC encoders the ME/DE represents more than 98% of the overall energy consumption. Moreover, in the overall ME/DE energy, up to 90% is related to the memory issues, and only 10% is related to effective computation. The two items to be concerned with: (1) off-chip memory communication to fetch the reference samples (45%) and (2) on-chip memory to keep stored the search window samples and to send them to the ME/DE processing core (45%). The main goal of this work is to jointly minimize the on-chip and off-chip energy consumption in order to reduce the overall energy related to the ME/DE on MVC. The memory hierarchy is composed of an on-chip video memory (which stores the entire search window), an on-chip memory gating control, and a partial results compressor. A search control unit is also proposed to exploit the search behavior to achieve further energy reduction. This work also aggregates to the memory hierarchy a low-complexity reference frame compressor. The experimental results proved that the proposed system accomplished the goal of the work of jointly minimizing the on-chip and off-chip energies. The RCDR provides off-chip energy savings of up to 68% when compared to state-of-the-art. the traditional MB-centered approach. The partial results compressor is able to reduce by 52% the off-chip memory communication to handle this RCDR penalty. When compared to techniques that do not access the entire search window, the proposed RCDR also achieve the best results in off-chip energy consumption due to the regular access pattern that allows lots of DDR burst reads (30% less off-chip energy consumption). Besides, the reference frame compressor is capable to improve by 2.6x the off-chip memory communication savings, along with negligible losses on MVC encoding performance. The on-chip video memory size required for the RCDR is up to 74% smaller than the MB-centered Level C approaches. On top of that, the power-gating control is capable to save 82% of leakage energy. The dynamic energy is treated due to the candidate merging technique, with savings of more than 65%. Due to the jointly off-chip communication and on-chip storage energy savings, the proposed memory hierarchy system is able to meet the MVC constraints for the ME/DE processing.

Keywords: Multiview Video Coding, 3D-Video, Low-Power Design, On-Chip Video Memory, Memory Hierarch, Energy Efficient, Motion Estimation, Disparity Estimation.

RESUMO

Esta dissertação de mestrado propõe uma hierarquia de memória para a Estimação de Movimento e de Disparidade (ME/DE) centrada nas referências da codificação, estratégia chamada de *Reference-Centered Data Reuse* (RCDR), com foco em redução de energia em codificadores de vídeo multivistas (MVC - *Multiview Video Coding*). Nos codificadores MVC, a ME/DE é responsável por praticamente 98% do consumo total de energia. Além disso, até 90% desta energia está relacionada com a memória do codificador: (a) acessos à memória externa para a busca das referências da ME/DE (45%) e (b) memória interna (*cache*) para manter armazenadas as amostras da área de busca e enviá-las para serem processadas pela ME/DE (45%). O principal objetivo deste trabalho é minimizar de maneira conjunta a energia consumida pelo módulo de ME/DE com relação às memórias externa e interna necessárias para a codificação MVC. A hierarquia de memória é composta por uma memória interna (a qual armazena a área de busca inteira), um controle dinâmico para a estratégia de *power-gating* da memória interna e um compressor de resultados parciais. Um controle de buscas foi proposto para explorar o comportamento da busca com o objetivo de atingir ainda mais reduções de energia. Além disso, este trabalho também agrega à hierarquia de memória um compressor de quadros de referência de baixa complexidade. A estratégia RCDR provê reduções de até 68% no consumo de energia quando comparada com estratégias estado-da-arte que são centradas no bloco atual da codificação. O compressor de resultados parciais é capaz de reduzir em 52% a comunicação com memória externa necessária para o armazenamento desses elementos. Quando comparada a técnicas de reuso de dados que não acessam toda área de busca, a estratégia RCDR também atinge os melhores resultados em consumo de energia, visto que acessos regulares a memórias externas DDR são energeticamente mais eficientes. O compressor de quadros de referência reduz ainda mais o número de acessos a memória externa (2,6 vezes menos acessos), aliando isso a perdas insignificantes na eficiência da codificação MVC. A memória interna requerida pela estratégia RCDR é até 74% menor do que estratégias centradas no bloco atual, como Level C. Além disso, o controle dinâmico para a técnica de *power-gating* provê reduções de até 82% na energia estática, o que é o melhor resultado entre os trabalho relacionados. A energia dinâmica é tratada pela técnica de união dos blocos candidatos, atingindo ganhos de mais de 65%. Considerando as reduções de consumo de energia atingidas pelas técnicas propostas neste trabalho, conclui-se que o sistema de hierarquia de memória proposto nesta dissertação atinge seu objetivo de atender às restrições impostas pela codificação MVC, no que se refere ao processamento do módulo de ME/DE.

Palavras-Chave: Codificação de Vídeos Multivistas, Vídeos 3D, Projeto de Baixa-Energia, Memória Interna, Hierarquia de Memória, Eficiência Energética, Estimação de Movimento, Estimação de Disparidade.

1 INTRODUCTION

The increasing demand for immersive multimedia systems has driven the popularization of the 3D video technology that is present in a wide range of applications like teleconference, automotive infotainment, cinema and personal 3D mobile cameras or cellphones. The 3D videos are based on the multiview concept (MERKLE, SMOLIC, *et al.*, 2007) where multiple independent cameras record the same 3D scene from different observation points (viewpoints). Each independent video is called *view* and the composition of multiple views is exploited by the 3D application to provide the depth perception of the objects (SMOLIC, MUELLER, *et al.*, 2007). The multiple video streams represent a huge amount of data that must be processed and encoded before their storage or transmission.

Some case studies proved the feasibility of multiview video recorders in 3D-camcoders and 3D-mobile phones devices (SHARP, 2011) (FUJIFILM, 2010). Due to the battery-powered nature of such mobile devices, the energy consumption must be as small as possible to enable a good battery usage together with multiview video handling. This way, all the processing and encoding on the multiview videos must be as efficient as possible to meet the energy constraints.

The Multiview Video Encoding (MVC) (JVT TEAM, 2009) provides from 20% up to 50% more coding efficiency when compared to a simple H.264/AVC simulcast approach. These gains are provided due to the innovations of the MVC which exploit the inter-view correlation between frames of neighboring views. This innovative tool is called Inter-View Prediction and uses the Disparity Estimation (DE) as the main module. However, it results in significant increase in the encoding complexity and energy consumption. Along with the Motion Estimation (ME), which is the DE similar step in the Inter-Frame Prediction, the DE represents more than 90% of the MVC encoder energy consumption (ZATT, SHAFIQUE, *et al.*, 2011). For this reason, the ME/DE is the optimization target for reducing the energy consumption in the MVC encoders.

The ME/DE goal is to search for the best representation for the current block (block that is currently being processed) in one or more reference frames (already coded frames) (RICHARDSON, 2003). When the best match is found, a motion/disparity vector that indicates the position of the best match inside the reference frame is generated, so the decoder is able to reassemble the block on the other side. The search is performed within a Search Window (SW) by using a search algorithm. There are several different algorithms for this task, which can be divided into two groups: exhaustive and heuristic. The exhaustive option is the Full Search (FS) (KUHN, 1999) which analyzes all possible candidate blocks to find the best match. As alternative, the heuristic algorithms use some greedy local choices to predict the motion/disparity direction of the image block. The TZ Search is one example of a heuristic algorithm for

ME/DE that presents speed-up results of 23x when compared against the FS, with minimum quality drops (XIU-LI, SHENG-KUI e CAN-HUI, 2010). This algorithm is implemented in the MVC reference software (called JMVC).

The most part of the energy consumption related to the ME/DE processing is due to the fetching of the search window samples from the off-chip memory and their corresponding on-chip storage. Both off/on-chip energies represent more than 90% of the overall ME/DE energy consumption (45% for the on-chip memory storage and 45% for the off-chip memory communication) (ZATT, SHAFIQUE, *et al.*, 2011). The search window for disparity search must typically cover a range of [-96,+96) in both directions, horizontal and vertical respectively (JVT TEAM, 2006). This amount of reference samples must be accessed for each current block in every frame, of every view on the multiview video. With the current video coding scaling demand, where (a) higher resolutions (up to 4K), (b) higher frame rates (up to 120 frames per second) and (c) a larger number of cameras (from 4 up to 8) are required, the memory requirements and its consequences to the overall MVC encoder energy demand have been increased.

During the past years, several works exploit the data locality of the ME/DE input reference samples in order to store part of them on-chip and, this way, they can be reused. Several works already proposed data reuse solutions for the non-MVC H.264/AVC, like (TUAN, CHANG e JEN, 2002) (CHEN, HUANG, *et al.*, 2006). The MVC encoding is the target in (TSUNG, DING, *et al.*, 2007), where more sophisticated approaches were designed by using intensive statistical analysis as the basis for their proposals. Some of these strategies were implemented as memory hierarchies of such hardware architectures to design efficient memory solutions for ME/DE (ZATT, SHAFIQUE, *et al.*, 2011) (ZATT, SHAFIQUE, *et al.*, 2011) (SHAFIQUE, ZATT, *et al.*, 2012). From a different perspective, some works aim to reduce the memory overhead by compressing the data before the MVC encoder saves the reconstructed samples in the off-chip memory (for future ME/DE) (MA e SEGALL, 2011) (WANG, CHANDA, *et al.*, 2012) (GUPTE, AMRUTUR, *et al.*, 2011) (SONG, ZHOU, *et al.*, 2010) (SILVEIRA, GRELLERT, *et al.*, 2012) . These works try to design efficient coding algorithms with as minimum computational effort as possible. The goal is to exchange off-chip memory bandwidth (responsible for 45% of the encoder energy) by computation overhead (around 10% of the entire energy).

The related approaches' results do not scale in a good way with the scaling for multiview videos. As higher is the number of views as higher is the number of the required ME/DE and more reference frames are inserted to be searched. None of the related works exploit the locality of the memory accesses in the reference frame level. The adopted ME/DE data reuse strategies are centered on the current block processing order, so one reference frame must be accessed many times and at different time instants, making it harder this level of data reuse. This work proposes a different schedule for the ME/DE operations, which is based on the accessed search window that is currently present in the on-chip memory.

Another point is related to the wasting of energy of unprocessed samples of the search window when heuristic search algorithms are used, like the TZ Search. Considering this algorithm, on average 80% of the samples in the search window are not analyzed in the TZ Search execution and, this way, the energy to access and store these samples is wasted (ZATT, SHAFIQUE, *et al.*, 2011). Some works reduce these penalties by a predictive prefetching for the most probable samples considering the history of previous accesses for the past blocks. However, this approach implies in an irregular off-chip memory access which complicates the control and does not match the regular burst access for the off-chip DDR memories (JEDEC, 2010). This work allies a regular pattern of access, by fetching the entire search window, and implements an on-chip memory power management to reduce the supply-voltage of the non-accessed regions of the search window to save on-chip memory energy.

Regarding the reference frame compression techniques, most of them fail to achieve good savings in the off-chip memory communication along with negligible losses in the encoder efficiency. This work aims to exploit lossless, lossy and adaptive solutions to meet with this tradeoff: high savings in the off-chip memory bandwidth and minimum losses in the MVC encoder performance.

The main challenge in this type of design is to *jointly minimize the on-chip and off-chip energy consumption in order to reduce the overall energy related to ME/DE on MVC.*

1.1 Contributions of this Master Thesis

This work proposes a reference-centered memory hierarchy for ME/DE on MVC targeting low-energy consumption at both on-chip storage and off-chip memory access. The memory hierarchy is composed of an on-chip video memory, an on-chip memory power gating control, and a partial results compressor. A search control unit is also proposed to exploit the search behavior to achieve further energy reduction. This thesis also integrates to the memory hierarchy a low-complexity reference frame compressor. The goal is to improve the savings by reducing the representation of the reconstructed samples before they are saved on the off-chip memory. The contributions of this Master Thesis are summarized as follows:

- **Reference-Centered Memory Hierarchy:** it employs a Reference-Centered Data Reuse (RCDR) scheme. It makes the reference frames the center of processing order to avoid search window retransmission and to eliminate the need to simultaneously store on-chip multiple search window. A memory access scheduling and an energy-efficient on-chip memory organization are proposed.
- **Statistic-Based Partial Results Compressor:** the RCDR strategy implies out-of-order processing considering the blocks perspective, since it is not guaranteed that one block of the video will be completely processed after the ME/DE operation. This way, the partial results (motion/disparity vectors and SAD) should be stored to be used for the next ME/DE over the same block. Statistically defined non-uniform quantization and Huffman coding are employed for partial results compression.
- **On-chip Video Memory:** the on-chip memory is organized in multiple SRAM banks featuring line-level power gating capability. At run-time, search window regions that are statistically less likely to be used are power-gated. Furthermore, the candidate blocks coding order is rearranged to minimize on-chip memory line switching and, consequently, the dynamic energy consumption.
- **Low-Complexity Intra-Based Reference Frame Compressor:** this compressor exploits the regular off-chip access pattern allowed by the RCDR strategy. The proposed low complexity compression algorithm is based on a simplified instance of the intra prediction defined in the H.264/AVC. The idea is use the encoder knowledge to just get the best intra mode for both I4MB and I16MB modes (RICHARDSON, 2003) that have been already processed by the MVC

encoder to code the reconstructed block before it is sent to the off-chip memory. Besides, non-linear quantization is applied in different strengths depending on the image region characteristics to reduce the losses and meet with high off-chip bandwidth reduction.

1.2 Text Organization

Chapter 2 presents the MVC background and points some MVC challenges related to the memory issues used as motivation for this work. Chapter 3 discusses the already proposed energy-efficient solutions for MVC encoding regarding the memory issues. Chapter 4 describes the proposed memory hierarchy system targeting energy-efficient MVC encoders. Chapter 5 presents the experimental results and comparison with the state-of-the-art works presented in the literature. Finally, Chapter 6 will conclude the work and will point some future directions for research.

2 MULTIVIEW VIDEO CODING BACKGROUND

During the past years, there was an intense research effort to provide even more realism for the current and the next generation multimedia applications. This effort has had as results the gradual popularization of high realism devices, like 3D television sets, which are definitely available for the common costumers for affordable prices. Other costumer devices like digital cameras, video games and cell phones already support the 3D video processing.

The digital video streams that support these 3D applications are not the usual ones taken by one single camera. Instead, the bases are videos that are captured by multiple cameras disposed at different observation points (viewpoints). Each individual video is called a *view* and the entire set of views composes the so called *multiview video*. One of the key challenges is how to transmit and store this kind of stream, since they require huge amount of bits to be represented. In this context, the Multiview Video Coding (MVC) acts in order to reduce the representation of these videos by exploiting the data redundancies (like objects that remain visible during several frames or homogeneous scene backgrounds).

This chapter initially discusses details of the overall multiview video systems, from the encoding part to the transmission systems and the final applications. Then, the multiview video characteristics are presented. The focus is to introduce the concepts of redundancies to understand how the video encoders work with the goal of exploiting them to reduce the data required for video representation. The basic concepts in multiview video coding are presented, and the main coding tools are described. Further, the Motion and Disparity Estimation, the main focus of this Master Thesis, are explained in more details. The Intra-Frame Prediction is then presented. Finally, this chapter points the main memory-related challenges in the multiview video encoders and set the contribution aimed in this work within these challenges.

2.1 Multiview Video Encoding System

Figure 2.1 presents the end-to-end Multiview Video Coding (MVC) System for some final applications (CHEN, WANG, *et al.*, 2009).

In this illustration, a multiview video is first captured and then encoded by a multiview video encoder (MVC encoder). A server transmits the coded bitstream(s) to different clients with different capabilities, possibly through media gateways. As the final stage, coded video is decoded and rendered with different meanings according to the application scenario and capabilities of the receiver. The scenario (a) from Figure 2.1 represents the so called free-viewpoint television, (b) is the wide view angle television, (c) corresponds to the narrow view angle television and (d) represents the 3D applications. All these scenarios combine multi video streams as input (multiview videos). Besides, multiview videos are compliant to an ordinary singleview video decoder. This is the case of scenario (e), where only one view (called base view) is decoded and exhibited from a common 2D display.

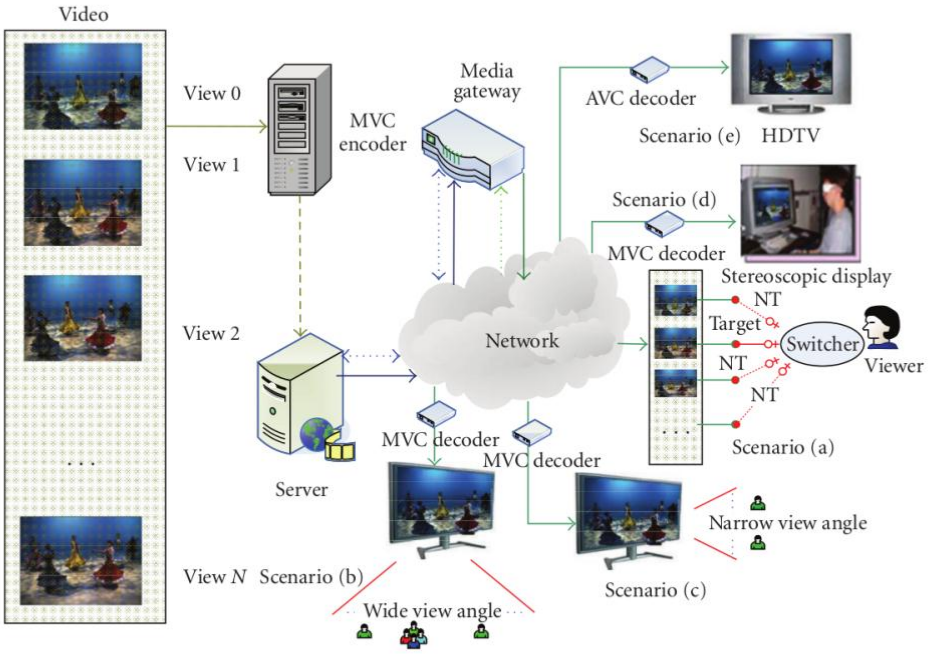


Figure 2.1: MVC system architecture. (CHEN, WANG, *et al.*, 2009)

The next sections initially describe the MVC final applications presented in Figure 2.1. Then, the MVC requirements and standardization are presented. Finally, details about MVC extension of H.264/AVC video coding standard are provided.

2.1.1 Multiview Scenarios and Applications

The primary usage for multiview video is to support 3D video applications, where 3D depth perception of the visual scene is provided by a 3D technology display system. There is a wide range of 3D displays systems (KONRAD e HALLE, 2007) including the classic stereo system that requires special-purpose glasses to more sophisticated multiview autostereoscopic displays that do not require the use of glasses (DODGSON, 2005). The stereo systems only require two views, where the left-eye is presented to the viewers' left eye, and the right-eye view is presented to the viewers' right eye. The 3D displays and the glasses ensure that the appropriate view is viewed by the correspondent human eye. This is accomplished with either passive polarization or active shutter techniques. The multiview displays have much greater throughput restrictions relative to conventional stereo displays in order to support a given image resolution, since 3D is achieve by essentially emitting multiple complete video sample arrays in order to form view-dependent pictures. Such displays can be implemented, for instance, using conventional high-resolution displays and parallax barriers; other technologies include lenticular overlay sheets and holographic screens. Each view-dependent video can be thought of as emitting a small number of light rays in a set of discrete viewing directions (typically between eight and a few dozen for an autostereoscopic display). Generally these directions are distributed in a horizontal plane, such that parallax effects are limited to the horizontal motion of the observer (VETRO, WIEGAND e SULLIVAN, 2011).

Another goal of multiview video is to enable free-viewpoint video (SMOLIC e KAUFF, 2005). In this case, the viewer can interactively change the viewpoint and the view direction of the 3D scene. Each output view can either be one of the input views or a virtual view synthesized from a smaller set of multiview inputs and other data that assists in the view synthesis process (such as depth maps (YAN, YANG, *et al.*, 2011)). With such a system, viewers can freely navigate through the different viewpoints of the scene, within a range inside the multiple cameras. This kind of application can be implemented by using simple 2D conventional displays. However, 3D displays for free-viewpoint systems could also be considered. It is possible to see the use of this functionality in broadcast production environments, or to change the viewpoint of a sports scene in order to catch a better angle of the play. Such functionality may also be of interest of gaming, education, surveillance, and sightseeing applications. As the final step, it can imagine providing this interactively capability directly to the home viewer.

Another important application of multiview videos is to support immersive teleconference applications. Beyond the advantages provided by the 3D displays systems, it is reasonable to use this technology to enable a more realistic communication experience when motion parallax is supported. Motion parallax is caused by the change in the appearance of a scene when the viewers shift their positions, causing the revealing of occluded scene contents.

For all of the above scenarios, the storage and transmission capacity requirements are significantly increased. Consequently, there is a strong need for multiview video compression techniques. These requirements, besides the current standardization on multiview video compression are discussed as follows.

2.1.2 Multiview Encoding Requirements and Standardization

The central requirement for most video coding designs is high compression efficiency. In the specific case of MVC this means a significant gain when compared to independent compression of each view. The compression efficiency of video encoders measures the tradeoff between cost (in terms of bitrate) and benefit (in terms of video quality) (SULLIVAN e WIEGAND, 1998). However, the compression efficiency is not the only criterion under consideration for the video encoder standardization. General further requirements for video encoder capabilities are also required, such as minimum resource consumption (memory, processing energy), low delay, high performance for a range of video resolutions, color sampling structures, and bit depth precisions.

Some requirements are specific related to MVC. Besides the temporal random access, it is also required the *disparity random access*. Together both ensure that any image can be accessed, decoded, and displayed by starting the decoder at a random access point and decoding a relatively small quantity of data which that image may depend. *View scalability* is also fundamental for multiview video encoders. It is related to the ability of a decoder to access a portion of the bitstream to output a subset of the encoded views (VETRO, WIEGAND e SULLIVAN, 2011). Another important requirement is the *backward compatibility*, which means that a subset of the MVC bitstream corresponding to one “base view” needs to be decodable by an ordinary (non-MVC) decoder, and the other data representing other views should be encoded in such a way that will not affect that base view decoding capability. The ability of an encoder/decoder to provide parallel processing was also required to enable practical implementation and to manage processing resources effectively. Furthermore, for ease of implementation, it was also highly desirable for the MVC design to have as many common elements as possible with an ordinary non-MVC system.

2.1.3 H.264/MVC Standard

The H.264/AVC video coding standard (JVT TEAM, 2003) was defined by a jointly group composed of experts from two important standardization companies: the ISO/IEC (International Organization for Standardization) (ISO, 2012) and the ITU-T (International Telecommunication Union - Telecommunication Sector). This new group was called JVT (Joint Video Team).

The Multiview Video Coding (JVT TEAM, 2009) was proposed in the Annex H of the H.264/AVC standard as the extension to efficiently deal with multiview videos due to the current demand on immersive multimedia applications that support this new kind of video streaming. Besides extended syntax elements, the MVC extension innovates due to the exploitation of the inter-view (also called disparity) correlations between frames of different views, by inserting the inter-view prediction. In doing so, the MVC encoders is able to achieve from 25% to 50% more compression gains compared to the simulcast approach, where the views are encoded as independent videos by non-MVC H.264/AVC encoders (MERKLE, SMOLIC, *et al.*, 2007). Moreover, quality gains of more than 3dB is also achieve. However, the computational effort inserted by the inter-view prediction was increased from 10 to 19 times in the MVC encoders. Besides, other design challenges emerge, like impressive memory and energy constraints.

The following sections will basis the digital video characteristics that are important for the video compressors. Then, the multiview video coding basis is explained from the generic MVC encoder description to the detailed discussion of the main focus of this work: the inter-frame and inter-view prediction.

2.2 Digital Video Characteristics

A digital video is a sequence of static images (called *frames*) that gives the motion sensation when exhibited in a certain rate. Typically, this frame exhibition rate should be between 24 and 30 frames per second to provide smoothly motion for the human visual system (RICHARDSON, 2003). However, nowadays the frame requirements are scaling due to the new demand for high realism. This way, elevated frame rates as 60 and 120 frames per second are already needed for state-of-the-art applications.

Each frame of the video is represented by a pixel matrix of dimensions *W* (width) and *H* (height). Each pixel stores the color and the luminosity of that position. In this sense, there are a few well-known color spaces to numerically represent one pixel. The most used color space is the RGB, which splits the pixel information in three color channels: red (R), green (G) and blue (B). The RGB color space is widely used in televisions, monitors and digital cameras. However, the RGB is not the preferred one for the video compression algorithms, since there is a compression high correlation between each color channel and, for this reason, it is not possible to apply different coding techniques to each component individually (RICHARDSON, 2003).

Instead of using RGB, the video encoders use the YCbCr color space. In this alternative space, the pixel information is divided in *luminance* (Y), *blue chrominance* (Cb) and *red chrominance* (Cr). The luminance (also referred as luma) channel represents the light intensity (grayscale) of the image. The chrominance (or chroma) components express the color tones of the pixels. In this color space there is not a high correlation between the YCbCr components (RICHARDSON, 2003). So, they can be processed by different algorithms that are able to exploit its specific properties in order to achieve better compression rates.

This work will call each of luma and chroma components of one pixel as *sample*. Then, one pixel in the YCbCr color space will have one luma, one blue chroma and one red chroma sample associated with it.

The multiview videos are composed of several videos taken by cameras that observe the scene from different viewpoints. The multiview videos inserted another dimension on the digital video representation. Conceptually, a singleview video is a sequence of 2-dimensional matrices (using RGB or YCbCr color spaces, for example) that represents the video in the spatial and in the temporal domain. The multiview videos have one more dimension: called disparity, and it is represented by the several independent singleview streams (MERKLE, SMOLIC, *et al.*, 2007).

Multiview videos require a huge amount of bits to be represented. The video encoders aim to reduce this information by exploiting the data redundancies that are presented in the digital videos. The most important one is the temporal redundancy. This kind of correlation is related to the high similarity between two consecutive temporal frames. Since the frame rates are usually greater than 30 frames per second, there is a lot of repeated information that are presented in two neighbor frames (WIEGAND, SULLIVAN, *et al.*, 2003). The differences are generally caused by the different position of the objects in the scene, due to their motion. Even in this case, the objects generally are presented in the frame and just its position was changed. The video coding algorithms exploit this kind of redundancy by performing searches for best matches between regions of two or more neighbor frames. The goal is to identify the modulus and the direction of the motion.

Another important video characteristic is the homogeneity between neighbor pixels in the same frame. Regions like a scene background and a blue sky are examples of very homogeneous areas. This characteristic is called spatial redundancy and it is exploited in video encoders by basically try to copy the neighbor pixels to infer the other neighbor ones. The way that is performed this copy must respect the texture direction of the represented object.

Besides, some studies about the human visual system points that some characteristics the images, like high frequency regions, are not perceived by the human eye. This way, the elimination of these high frequency elements considerably increases the compression rates due to negligible losses in the perceptual quality. This kind of redundancy is classified as psycho-visual (GONZALEZ e WOODS, 2003) or as spatial (RICHARDSON, 2003), depending on the author.

The multiple views insert an additional video characteristic: the redundancies between frames from different views. In this case, the redundancy is not related to the temporal correlation, but to the multiple cameras that are observing the same scene at the same time instant, i. e., by the disparity correlation. This type of correlation is called disparity redundancy and it is generally exploited by video encoders by the same block matching process that is employed to deal with the temporal redundancies.

Figure 2.2 presents some frames of the VGA (640x480 pixels) test sequence *ballroom* to show examples of redundancies in a real video. The *ballroom* sequence is used in the standardized benchmark set used for the video coding community (JVT TEAM, 2006).

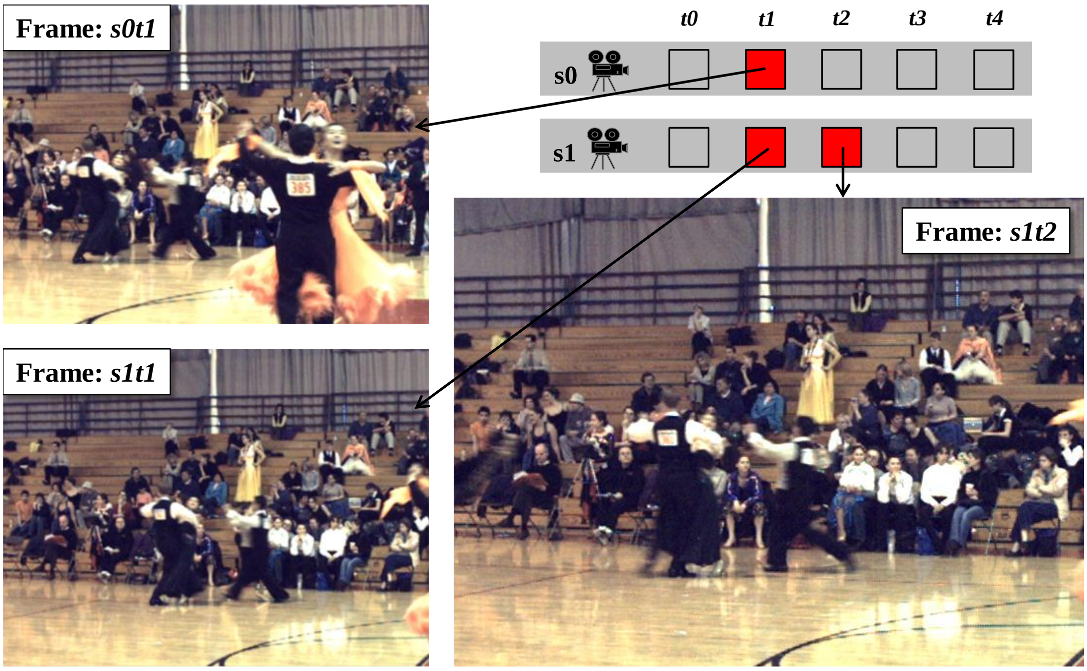


Figure 2.2: Redundancies in multiview videos.

In this whole Master Thesis, a frame will be indicated by the notation presented on Figure 2.2. For example, the frame "*s1t2*" indicates the frame of the view *s1* captured at the time *t2*. Taken the frame *s1t2*, it can be noted regions of the image that are spatially similar, like the black dancers clothes. As already presented, this correlation among neighboring pixels in the same frame is called spatial redundancy. Furthermore, if two temporal neighboring frames are taken (*s1t1* and *s1t2*, for example), the images are almost equals due their high temporal redundancy. The higher motions are caused by the dancers that are on the rightmost part of the frames. On the other perspective, the disparity estimation can be noted when the frames *s0t1* and *s1t1* are compared. In this case, the differences are not related to the motion, but to the different camera viewpoints. Also, lots of redundancies can be perceived.

2.3 Basic Concepts on Video Coding

As already explained in Section 2.1, the video compression is located in an extremely important role in the multiview video generation, transmission and reproduction system. This importance is increased when it is considered the scaling demand that is required for the new multimedia applications: frame resolution scaling, frame rate scaling and view number scaling. The multiview video treatment is unfeasible, at least at the current technology, if a non-compressed approach is used (MERKLE, SMOLIC, *et al.*, 2007).

The video compression (usually also called *video coding*) aims to reduce the data representation of the digital video by exploiting its existing redundancies. The main tradeoff of a video encoder is to ally high compression rates with minimal losses in visual video quality (SULLIVAN e WIEGAND, 1998). This tradeoff is commonly referred as *rate-distortion optimization* and it will be further explained.

The latest video coding standards adopt as default color space the YCbCr. This way, the digital video are composed of one luma samples matrix (Y) and two chroma samples matrices (Cb and Cr). These matrices are divided in basic unities called *blocks*, the basic coding unit of a video encoder. The coding algorithms will be applied individually for each block of the frame. The dimension of the block can vary in according with the encoder standard. The H.264/AVC standard defines a fixed-size block of 16x16 samples as the basic block size, called macroblock (MB). During the coding process, this macroblocks can be subdivided in smaller blocks (16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 are allowed) (WIEGAND, SULLIVAN, *et al.*, 2003).

The first basic technique to reduce digital video representation is to discard some samples that are not important for the human eye perception. In this sense, the less important sample channels in the YCbCr color space are the chroma ones (RICHARDSON, 2003), since it was studied that the video quality perception for the human eye is more related to the luma samples resolution matrix. This way, it is possible to reduce the resolution of the chroma channels with negligible losses in quality for the human visual system. These different relations between the luma and chroma dimensions are called *color formats*.

The most common formats are 4:4:4, 4:2:2 and 4:2:0. In the 4:4:4 format, each Cb and Cr sample are related to only one luma sample. This is a non-compressed scenario where chroma and luma resolutions are the same. The 4:2:2, 4:1:1 and 4:2:0 are called *subsampled* formats, where the chroma resolution is smaller than the luma resolution.

The latest video coding standard, like H.264/AVC, the adopted correlation between luma and chroma samples is 4:2:0. It means that the blue and red chroma matrices for one frame have their dimensions subsampled to (*W/2)x(H/2)*, while the luma resolution remains *WxH*. This way, this work will adopt this data format in all experimental analyses.

The basis scheme behind the latest video encoders is called *residual coding*. In this strategy, the goal is to analyze different ways to code a given block of pixels by using previously coded blocks of the video. These already coded blocks are taken as reference for the current block coding. This task is called *prediction* and it is the core of the residual encoders. The prediction process is responsible to search for the best way to represent the block that is being coded by using all already coded blocks as reference (in the past frames or in the same frame). As final result of the prediction step, two main elements are generated: (1) the *prediction block* that represents the best possible representation of the original block by using only the reference blocks and (2) the *prediction mode* that tells the way to use the reference pixels to generate the predicted block. The predicted block can be different when compared to the original one, so the simple discarding of these differences will result in losses in the pixel values. In order to deal with it, this difference is also generated and it is sent together with prediction mode (called *residual block*). In the decoder process, the predicted block is generated in an inverse way, by having the prediction mode. The original block can be remounted by adding the residual information with the predicted block.

The H.264/AVC standard (JVT TEAM, 2003), which is the adopted one for all experiments in this work, uses the residual encoding concepts. Besides, the next generation of video encoders also applies residual coding besides to new features to handle with even higher resolution videos (JCT-VC, 2012).

There are a large variety of techniques that were proposed to improve the compression rates in video encoders. In order to delimit the scope of this work, the following explanations will take as basis the techniques defined by the H.264/AVC encoder for multiview videos.

2.4 H.264/AVC Encoder for Multiview Videos

The Figure 2.3 presents the overall diagram of the H.264/AVC multiview video encoder with its main coding tools. The main modules of the encoding process are: Motion Estimation (ME) and Motion Compensation (MC), composing the Inter-Frame Prediction; the Disparity Estimation (DE) and the Disparity Compensation (DC), composing the Inter-View Prediction; the Intra-Frame Prediction; the Mode Decision; the Forward Transforms and the Forward Quantization (FT/FQ), which are responsible for the residual information processing; and, finally, the Entropy Coding.

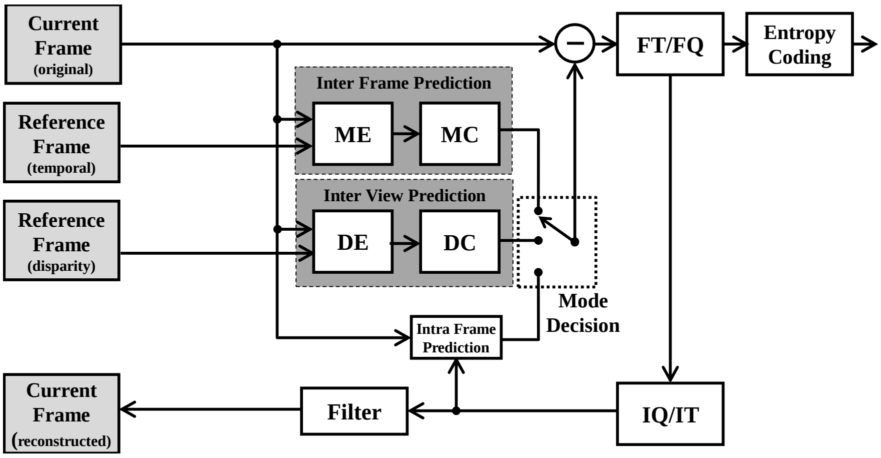


Figure 2.3: Overall H.264/AVC encoder block diagram for multiview videos.

Inside the encoder, there are a reconstruction path, composed of the Inverse Quantization and Inverse Transforms (IQ/IT), to deal with the mismatch between encoder and decoder. This issue will be discussed further. Besides, a Deblocking Filter is also used to smooth the borders between the blocks, increasing the subjective quality of the video.

As presented in Figure 2.3, the prediction process in the H.264/AVC is divided in three different modules: (1) the Intra-Frame Prediction, (2) the Inter-Frame Prediction (ME and MC) and (3) the Inter-View Prediction (DE and DC). As already explained in the Section 2.3, the prediction process is responsible to find the prediction mode that generates the most similar predicted block to the original block.

The Intra-Frame Prediction aims to exploit the spatial redundancy. Due to several predefined copy modes, the intra prediction predicts the current block by copying the already coded neighbor samples in several predefined directions. The main profile of the H.264/AVC defines the intra prediction for 16x16 and 4x4 block sizes. There are 4 possible prediction modes for the 16x16 blocks and 9 possible modes for the 4x4 modes. The intra prediction process will be further detailed in the Section 2.6.

The Inter-Frame Prediction is the task that deals with the temporal redundancy and it is composed of two main subtasks: the Motion Estimation (ME) and the Motion Compensation (MC). The goal of the ME is to effectively search for the best match between the current block in one or more already coded temporal neighbor frames (called *reference frames*). Once the best match is found, the ME generates a *motion vector* and a *reference frame index* in order to indicate to the decoder in which reference frame the best match is and in which position it is located. The MC is the next step and it is responsible to access the reference frame memory to get the predicted block by using the ME outputs (motion vector and reference frame index). Finally, when the MC remounts the predicted block from the memory, the residual information is calculated and sent to the Forward Transforms and Forward Quantization to be properly coded.

The Inter-View Prediction is an innovation proposed by the Annex H of the H.264/AVC released in 2009 (JVT TEAM, 2009). The Inter-View Prediction is responsible to exploit the disparity correlations between neighboring frames of different views. The insertion of this coding tool increases the compression rates of multiview videos from 20% to 50%, according to the video characteristics. However, the computation effort of the overall encoder is increased in almost 19x (MERKLE, SMOLIC, *et al.*, 2007). The steps of this prediction module are the Disparity Estimation (DE) and the Disparity Compensation (DC) and they conceptually works in a similar way to the ME/MC steps in the Inter-Frame Prediction. The difference is that the references for the DE are the already coded disparity neighboring frames, differently from the ME, which searches in temporal neighboring frames. Although the ME and DE are conceptually similar, there are important characteristics that differ one from the other, since they are working with distinct redundancies: ME exploits the temporal redundancy, i. e., the motion of the objects, and the DE acts to eliminate the disparity redundancy and it needs to catch the objects displacements due to the different camera viewpoints (MERKLE, SMOLIC, *et al.*, 2007). These particular properties of Inter-Frame and Inter-View Predictions will be further discussed in Section 2.5.

The residual data is then processed by the Forward Transforms and Forward Quantization before they are packed in the final *bitstream* (coded video). These two modules compose the *residual treatment path*, which consists of a series of mathematical calculations that will prepare the residual blocks to be efficiently coded by the final encoding step, the Entropy Coder. First, the transforms will be applied to change the samples from the spatial domain to the frequencies domain (WIEGAND, SULLIVAN, *et al.*, 2003). The transformed block group the frequencies in a descending order. Then, the quantization acts on the transformed block to take the lower frequencies to zero (full compression) and to attenuate the higher frequencies to lower values. The quantization strength is controlled by an external parameter, called Quantization Parameter (QP). This is the encoding step that inserts losses in the coded video, since the quantization basically performs an integer division in the transformed residual samples. This way, the encoder is not able to recover the original value. The QP is the main external parameter to control the encoder efficiency and acts in the sense to control the losses and the compression rates of the entire encoder. As higher is the QP value, higher will be the compression rates and more perceived are going to be the losses. On the other hand, lower QP values will generate minor losses and worst compression rates. The quantization output is a sparse matrix with very low values that can be very efficiently compressed by the Entropy Encoder.

The three predicted blocks (Intra-Frame, Inter-Frame and Inter-View) are then analyzed by the Mode Decision. The goal now is to decide for the best mode that minimizes the efficiency tradeoff between video quality and final bitrate. This is a very complicated task and there are a lot of things that the Mode Decision needs to considerate to take a good choice. The reference software of the MVC, the JMVC (Joint

Multiview Video Coding Model), implements the technique called Rate-Distortion Optimization (RDO) (WIEGAND, SULLIVAN, *et al.*, 2003). The RDO is a very computation-intensive algorithm for Mode Decision that performs the full encoding process for all possible prediction modes. At the final, the RDO compares the rate-distortion costs and gets the best one. It means, in other words, that for each possible prediction mode the encoder need to execute the prediction, transforms, quantization and entropy encoding to generate its rate-distortion cost. For this reason, the RDO is extremely costly in terms of MVC encoder execution time.

The rate-distortion cost is the metric used by the RDO to choose the best prediction mode for a given block. Its mathematical definition is presented in the Equation (1), where D represents the distortion between the reconstructed block (after the quantization losses) and the original one, R is the output bitrate and λ is the *lagrangian* parameter which correctly weighted the tradeoff between distortion and bitrate depending on the target QP.

$$RD_{cost} = D + \lambda\, R \tag{1}$$

The prediction process is able to use different distortion metrics (also called similarity metrics) to measure the similarity degree between the candidate blocks and the current block. Also, these metrics are used in the RDO mode decision to calculate the distortion term (D in Equation (1)). These metrics will be detailed discussed in Section 2.5.

The mode decision accuracy directly impacts the final encoder performance in terms of quality losses and compression rates. The RDO is commonly used as superior borderline as the most efficiency and the most complex mode decision algorithm, since all possibilities are completely coded and reconstructed to get the final distortion and bitrate. For this same reason, the RDO provides the best rate-distortion results.

The criterion to define the quality of the generated video is extremely important to evaluate the encoder performance. However, there are a lot of issues that must be counted. The subjective visual analysis to determine the quality is very imprecise, since there are a lot of factors that affect the evaluation and some of them are inherently subjective. This kind of quality measurements is called *subjective metrics* (SESHADRINATHAN, SOUNDARARANJAN, *et al.*, 2010) and will not be discussed in this Master Thesis. This work adopts the *objective metrics* to evaluate the quality of the generated videos.

The objective quality is calculated by algorithms that compare the original video and the encoded video. This comparison is performed frame-by-frame by matching all pixels of the original frame with all most similar pixels of the encoded frame.

The PSNR (Peak-to-Signal Noise Ratio) is the most accepted objective quality metric by the video coding community (RICHARDSON, 2003). It is expressed in decibels (dB) and uses a logarithmic factor based on the MSE (Mean Squared Error) value of the reconstructed and original frames. The Equations (2) and (3) show the MSE and PSNR mathematical definitions, where R represents the reconstructed video frame and O is the original frame samples matrix.

$$MSE(x, y) = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (R_{i,j} - O_{i,j})^2 \tag{2}$$

$$PSNR = 10 \times \log_{10}\left(\frac{maxpixelvalue^2}{MSE}\right) \qquad (3)$$

In the Equation (3), *maxpixelvalue* represents the highest representable value of a sample. Generally, most of the H.264/AVC profiles use 8-bit representation per sample. This way, the *maxpixelvalue* is adopted as 255 in this work.

2.5 Motion and Disparity Estimation

The Motion Estimation (ME) and Disparity Estimation (DE) are the main core of Inter-Frame and Inter-View Prediction steps in the MVC encoders. ME is responsible to deal with the temporal correlations of neighboring frames from the same view, while the DE perform the same task but for the disparity correlations of frames from different views. Figure 2.4 presents a simple prediction structure to introduce some concepts, where *s0* and *s1* represent the views, {I, P, B} are the frame types in according to the possible prediction modes for each frame (WIEGAND e SULLIVAN, 2011), and the arrows represent the temporal/disparity dependencies.

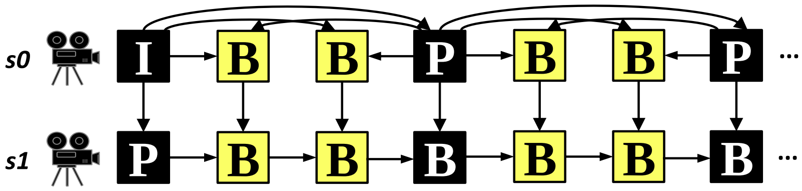


Figure 2.4: Illustration of inter-frame and inter-view prediction in MVC. (VETRO, WIEGAND e SULLIVAN, 2011)

Horizontal arrows (inside the same view) represent the ME operations. For instance, the frame *s0t1* has two ME coding dependencies: frames *s0t0* and *s0t3*. It means that these dependencies must be coded, reconstructed and stored on the reference frames memory before the frame *s0t1* is started to be coded. On the other hand, the vertical arrows in Figure 2.4 represent the DE operations. For example, the frame *s0t1* has both one ME dependency *s0t1* and one DE dependency with the frame *s1t0*.

The next sections present the basic concepts involved in the ME/DE processing, along with MVC typical prediction structures that specifies the way that ME/DE are disposed for a given number of camera views.

2.5.1 Basic Concepts

The basic concepts presented in this section are applied to both ME and DE processing flow. The conceptual difference between them is that ME exploits the temporal redundancies between neighbor frames in the same view, and DE processes frames in order to catch the camera disparity between frames of different views (generally captured in the same time instant). Figure 2.5 summarizes these concepts.

For each MB of the frame that is being processed (*current frame*), the ME/DE module is applied in according to the MVC encoder prediction structure. The goal is to find the *best match* of each block (called *current block*) of the current frame in one or more previously coded frames (called *reference frames*). The optimal best match corresponds to the block in the reference frames that provides the smallest possible rate-distortion cost in the final MVC encoding process. The optimal solution search is a very computation-intensive decision, since the entire MVC encoder process must be performed for all possible block matches in the reference frame (SULLIVAN e

WIEGAND, 1998). As a local decision, the block that minimizes the residual information (difference between the original MB and the predicted block) is always a very good candidate to be near to the optimal result (KUHN, 1999). Several low complexity metrics are used to measure the similarities between two blocks (they will be discussed on the Section 2.5.2). Using this metric, the current block is compared to blocks of the reference frame (called *candidate blocks*) and the most similar one is picked as the best match. As result, the ME/DE deliveries (1) a motion/disparity vector indicating the position of the best match and (2) a reference index that points to the specific reference frame that contains the best match (the search can be performed in more than one reference frame) (WIEGAND, SULLIVAN, *et al.*, 2003).

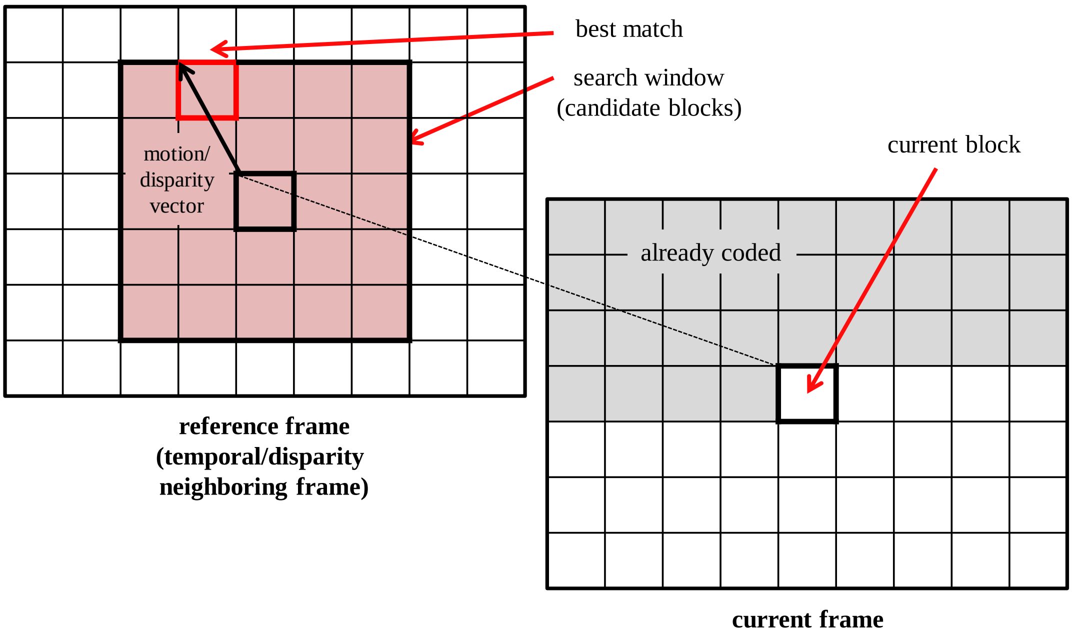


Figure 2.5: Motion and Disparity Estimation basic concepts.

Due to the high frame exhibition rate (30, 60 up to 120 frames per second), the best match tends to be found positions closed to the current MB. This way, the search on the entire frame blocks may represent so much effort since the best match is usually in the current MB neighborhood. This way, it is common to restrict the ME/DE search within a *search window*. Indeed, practically all ME/DE search algorithms use the search window to reduce the execution time and memory requirements (RICHARDSON, 2003). The DE has a different behavior, since the position displacement between each pair of camera is not guaranteed to be small and the best match may not be near the center of the search window. The H.264 MVC does not standardize a specific tool to deal with it, so larger search windows are required in the DE (at minimum 193x193 samples). Another standard-compliant solution is to use disparity vectors of previously MBs of the neighborhood to move the search window center to a more probable region for the best match. Moreover, several works evaluate the use of an external parameter, called Global Disparity Vector (GDV), which informs the average disparity between two view sequences (ZHEN, LIU, *et al.*, 2010).

The step after the ME/DE is the Motion and Disparity Compensation (MC/DC). In this task, the motion/disparity vector is used to access the reference frame memory to fetch the predicted block and, then, the MC/DC calculates the residual block and send it to the transforms and quantization.

The H.264/AVC standard inserted high complexity additional techniques to the ME/DE operation to improve the overall MVC encoder efficiency, like Variable Block Size (VBS-ME), quarter pixel precision and out-of-frame motion/disparity vector. These techniques will not be discussed because they are out of the scope of this work.

One important feature that is massively used in the MVC is the Multiple Reference Frames (MRF), where the ME/DE searches in more than one reference frame. The reference frames are organized in two lists (list 0 and list 1). Detailed description of the frame lists can be found in (WIEGAND, SULLIVAN, *et al.*, 2003). In the case of Figure 2.6, the same current frame is able to have references on five already coded reference frames (three past and two future frames, considering the exhibition order). It is massively exploited in the MVC encoders, since it is common to have past and future reference frames in the same view (processed by the ME) along with reference frames on the left and on the right camera view (encoded by the DE).

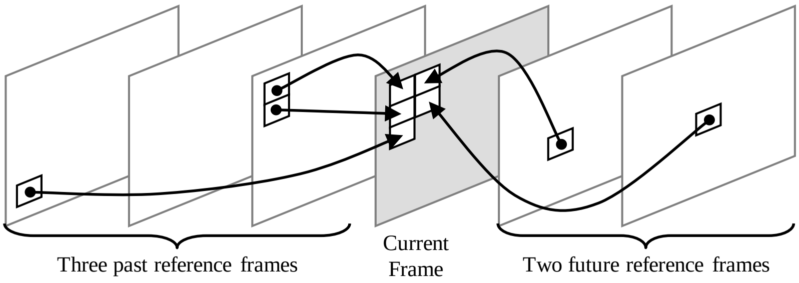


Figure 2.6: Multiple Reference Frame (MRF) example.

Another important concept is linked to the type of frames that ME/DE is able to be performed. The H.264/AVC standard classifies the frames of the multiview video in three different types (RICHARDSON, 2003).

- **I-Frame:** This type does not allow the ME/DE, since all MBs must be intra coded to not contain any encoding dependency with another frame. I-Frames are disposed in order to refresh the video dependencies to allow random access at certain points of the MVC decoding.
- **P-Frame**: This type allows the use at maximum one temporal/disparity reference for each MB. It means that the ME/DE is required for this type and, even searching in multiple reference frames, only one motion/disparity vector is generated for each MB or partitions/sub-partitions. Besides, the MBs of this frame type can also be intra coded.
- **B-Frame:** This type allows all possibilities of P-Frames, besides the bi-predictive estimation. The bi-prediction uses two motion/disparity vectors to predict the current MB. The predicted block is then generated by the arithmetic average operation between the samples of the reference blocks pointed by the two motion/disparity vectors.

2.5.2 Search Algorithms and Similarity Metrics

There are several algorithms that were proposed to perform the search for the best match in the reference frames. The MVC reference software, which is released and maintained by the JVT group (called JMVC) implements several search algorithms for the ME/DE processing on MVC (JVT TEAM, 2010).

The Full Search (FS) algorithm is the optimal solution that computes the exhaustive search for all candidate blocks (pixel by pixel) within the search window (KUHN, 1999). Due to the exhaustive search approach, the FS achieves the best rate-distortion results. However, the number of comparisons grows in a quadratic order with the search window increasing.

As an alternative solution, the JMVC also implements some fast search algorithms, which are greedy solutions that aim to direct their search due to heuristic calculations to derive the direction of the motion (or camera displacement). Among these algorithms, the best MVC encoder rate-distortion efficiency is achieved by the TZ Search algorithm. Compared to the FS algorithm, the TZ Search achieves a speed-up of 23x with negligible losses in the MVC encoder efficiency (XIU-LI, SHENG-KUI e CAN-HUI, 2010). The TZ Search is the algorithm used in this work as search engine for ME/DE.

Along with a good search algorithm, ME/DE processing also requires an accurate and simple similarity cost. There are several well-known metrics that can be used, like: Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD) and Sum of Absolute Transformed Differences (SATD). The mathematical definitions for each metric are presented in Equations (4), (5) and (6).

$$SAD(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|R_{i,j} - O_{i,j}\right| \tag{4}$$

$$SSD(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(R_{i,j} - O_{i,j}\right)^2 \tag{5}$$

$$SATD(x,y) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}HAD(|R - O|)_{i,j} \tag{6}$$

In the Equations (4), (5) and (6), *R* represents the predicted block and *O* is the original video block. Indeed, the *HAD* function is the Hadamard transform defined in the H.264/AVC (JVT TEAM, 2003).

2.5.3 Prediction Structures

As already mentioned, encoding and decoding each view of a multiview test data set separately can be done from non-MVC H.264/AVC codec. It would be a simple, but inefficient way to compress multiview videos, since the inter-view correlations are not exploited (MERKLE, SMOLIC, *et al.*, 2007).

In the temporal domain (same view), the MVC typically uses the concept of Hierarchical B-pictures (SCHWARZ, MARPE e WIEGAND , 2006). This type of prediction scheme takes benefit from the increased flexibility of H.264/AVC at frame level in comparison to other video coding standards through the availability of multiple reference frames technique (WIEGAND, SULLIVAN, *et al.*, 2003). A typical hierarchical prediction structure with three stages is depicted in Figure 2.7. The first frame of a video sequence is fully intra-coded (I-Frame description in Section 2.5.1), called anchor frame (black in Figure 2.7). Frames of this type are encoded in regular intervals to refresh the dependencies and to allow random access. An anchor frame and all frames that are temporally located between the anchor frame and the previous anchor frame are considered to build a Group of Pictures (GOP), as illustrated in Figure 2.7 for a GOP of eight frames.

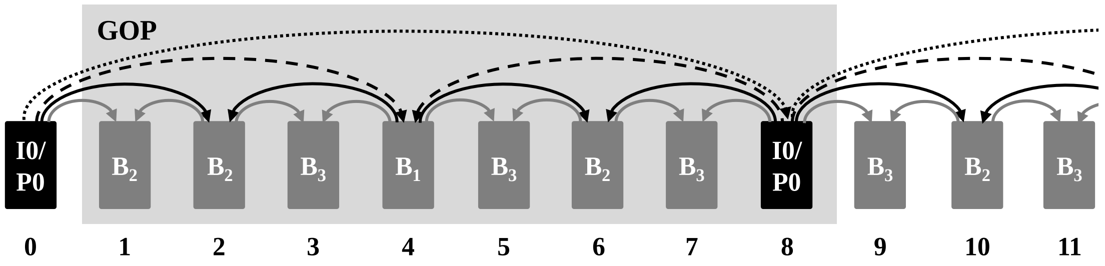
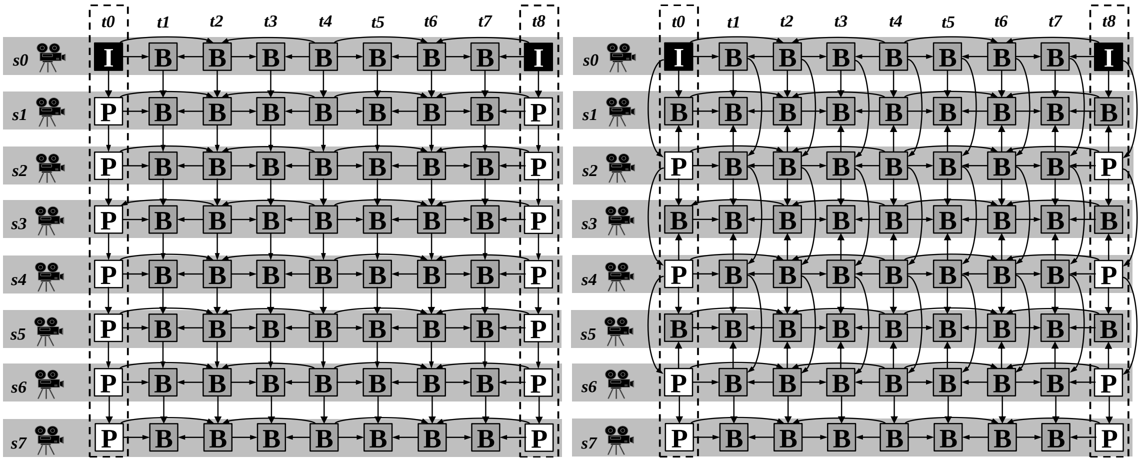


Figure 2.7: Hierarchical prediction structures for temporal prediction. (CHEN, WANG, *et al.*, 2009)

The temporal levels of the prediction structure are denoted by the indices [I_i, P_i, B_i], where i={0, 1, 2, 3} for GOP size of eight frames, in the Figure 2.7. The hierarchical prediction fashion ensures that all frame are predicted by using only frames of the same or a higher temporal hierarchy level as references, to support several temporal scalability levels (MERKLE, SMOLIC, *et al.*, 2007). As depicted in Figure 2.7, the B frames of such a hierarchical structure are typically predicted by using the two nearest frames of the next higher temporal level as references. Regarding the coding order, this prediction structure implies the constraint that the reference frames must be encoded before the current frame, so the coding order is different from the exhibition order.

The concept of hierarchical B frames can easily be applied to multiview video sequences as illustrated in Figure 2.8 for a sequence taken by 8 cameras and GOP length equals to 8: S_n represents the individual view sequences and T_n the time stamps. Then, two different approaches are presented: the "IPP" prediction structure, where the anchor and non-anchor frames have at maximum one view dependency (Figure 2.8a), and the "IBP" structure that defines an interlaced approach between neighboring views, even views have one disparity dependency (except from view S_0) and odd views are 2-view dependent (Figure 2.8b).



(a) IPP Structure

(b) IBP Structure

Figure 2.8: Typical MVC Prediction Structures: (a) "IPP" and (b) "IBP".

The "IBP" structure improves the disparity exploitation by inserting more inter-view dependencies (DE searches) in the MVC processing. It can be noted that in both IPP and IBP structures the anchor frames and the non-anchor frames have the same inter-view dependencies. For simplicity, such works usually avoid the DE dependencies for non-anchor frames. However, this work adopts the IPP and IBP prediction structures as they are presented in the Figure 2.8, with inter-view dependencies on both anchor and non-anchor frames.

2.6 Intra-Frame Prediction

The Intra-Frame Prediction is the MVC encoding tool responsible to exploit the spatial correlation inside a frame. This prediction is based on the previously coded samples in the same frame, specifically the upper and left neighboring samples related to the current block. The intra prediction for luma samples can be applied to 4x4 or 16x16 blocks (the entire MB). There are nine different modes of intra prediction for 4x4 blocks and four modes when the 16x16 blocks are targeted (RICHARDSON, 2003). This module is presented on both MVC encoder/decoder. Furthermore, the intra prediction is an innovation of the H.264/AVC standard to deal with the spatial dependencies inside the same frame.

Figure 2.9 presents an example of the neighboring samples that are used to process the intra prediction step. The gray part shows a 4x4 luma block that is being processed. The leftmost and upmost white samples (from “A” to “M”) were already coded and reconstructed by the encoding loop before the current block starts to be processed. These are the samples used as references for the intra prediction. The samples from “a” to “p” correspond to the predicted block, which is the intra prediction result.

M	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
I	**a**	**b**	**c**	**d**				
J	**e**	**f**	**g**	**h**				
K	**i**	**j**	**k**	**l**				
L	**m**	**n**	**o**	**p**				

Figure 2.9: Samples identification for the intra-frame prediction .

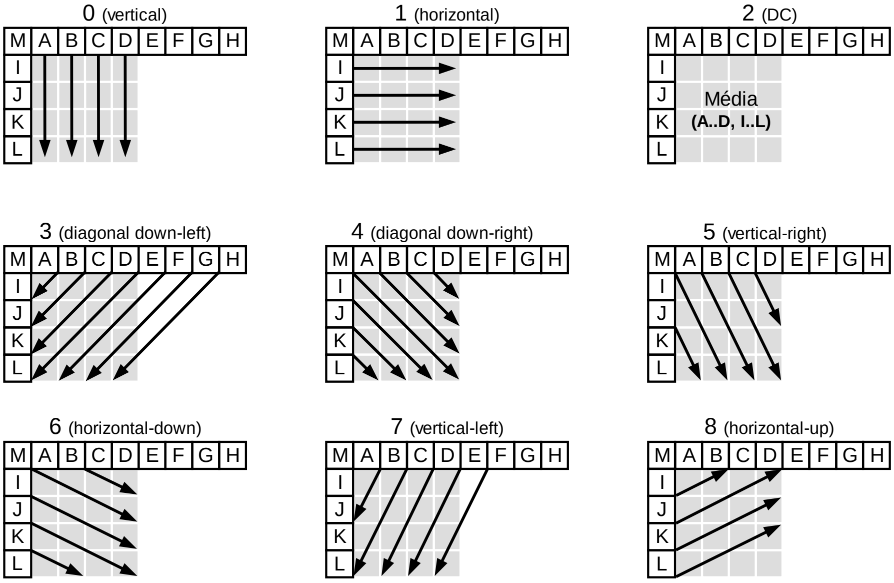


Figure 2.10: Luminance 4x4 block intra prediction modes.

Figure 2.10 presents the nine possible encoding types in the intra prediction modules for the 4x4 luma blocks. Modes 0 and 1 perform a simple extrapolation (copy) either vertical or the horizontal pixel border to all block positions. The DC mode (2) calculates the average value of the neighboring pixels and copies the result to all positions of the 4x4 block. The rest of the modes (3 to 8) perform weighted averages of the border samples, in according to the arrow direction of Figure 2.10.

As example, if it is considered the intra prediction for 4x4 luma blocks using the mode 4 of Figure 2.10, then the predicted sample calculation of the position “d” of Figure 2.9 is given by the Equation (7). On the other side, the formula that generates the value of “a”, “f”, “k” and “p” is provided by Equation (8).

$$d = round\left[\frac{B + 2C + D}{4}\right] \tag{7}$$

$$a, f, k, p = round\left[\frac{I + 2M + A}{4}\right] \tag{8}$$

As already mentioned, the intra prediction can also happen over the whole luma MB, i. e., 16x16 blocks. In this case, there are four possible intra prediction modes that can be applied, as can be seen in Figure 2.11.

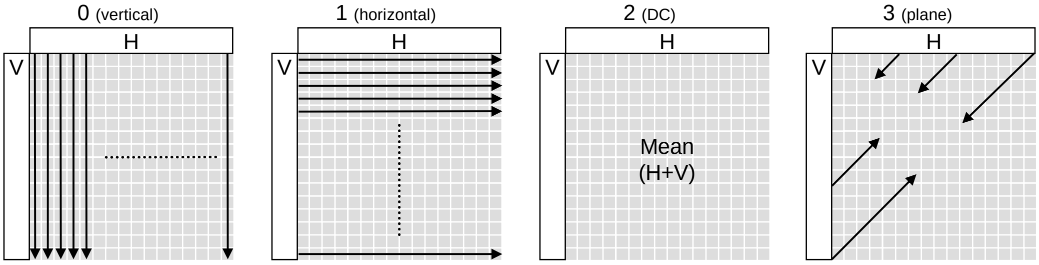


Figure 2.11: Four possible intra prediction modes for luma 16x16 blocks.

The chroma intra prediction is directly performed over 8x8 blocks and uses four different prediction modes, but the two chroma channels use always the same mode. The prediction modes for chroma are very similar than those presented on Figure 2.11, except by the mode indexing. In the chroma intra prediction scope, the DC mode is the zero, the horizontal mode is the one, vertical mode is the two and the plane mode is the number three (RICHARDSON, 2003).

The intra prediction itself is not the focus of this work, since there is no MVC memory restrictions involved on its processing. However, the proposed technique that aims to compress the reference frames before they are stored in the off-chip memory uses the MVC encoder knowledge of the intra prediction step to avoid computation and to reduce the data transmission between MVC encoder and DDR off-chip memory. In doing so, the Section 4.5 will require the intra prediction concepts that were explained in this section.

2.7 Memory Challenges on Multiview Video Coding

Memory-related issue has been demonstrated to be the most energy spending part in the most recent digital video encoders (ZATT, SHAFIQUE, *et al.*, 2011). Specially, the new demand for immersive/3D applications and the consequently development of MVC codecs requires even more data traffic with the system memory. Besides, the 3D-camcoders and 3D video processing cores that are present on mobile devices proves the need of MVC encoders embedded in the nowadays mobile devices. In doing so, energy-efficient techniques to address this memory energy challenge on MVC are required.

Figure 2.12 presents the energy analysis of the MVC encoding for the *ballroom* sequence for various search window sizes (ZATT, SHAFIQUE, *et al.*, 2011). The energy consumption (y-axis) is presented normalized to the highest search window related results, 257x257 size, considering five different search window sizes (x-axis).

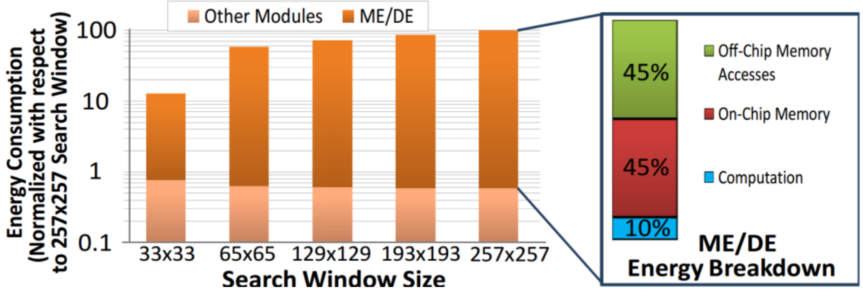


Figure 2.12: MVC Energy Breakdown (ZATT, SHAFIQUE, *et al.*, 2011).

It can be seen that the most energy consuming step is the ME/DE, representing 98% of the total MVC encoding energy. Besides the comparisons computations (SAD, SSD or SATD), the increased energy is mainly due to the enormous memory accesses to perform the memory-intensive block-matching process in ME/DE. The energy consumption due to the ME/DE processing is 90% occasioned by the on-chip storage and off-chip memory accesses (only around 10% is from the effective computation). As the non-MVC encoding already had the energy the memory as the energy bottleneck, as evaluated in (YANG, WOLF e VIJAYKRISHNAN, 2005), the MVC inserts the DE step that results in a significant increase in the memory energy consumption, because, typically larger search windows are deployed for DE. Therefore, it can be concluded that *there is a dire need to reduce energy of the on-chip memory (that stores the search window) and off-chip memory accesses in ME/DE in order to perform MVC encoding on battery-powered devices.*

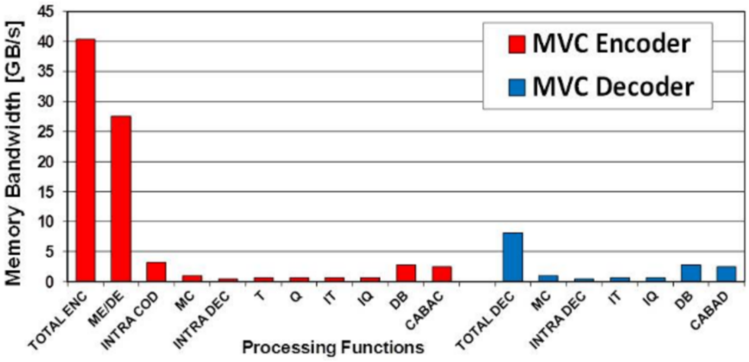


Figure 2.13: MVC Encoder/Decoder memory bandwidth requirements (SHAFIQUE, ZATT, *et al.*, 2012).

ME/DE requires a lot of off-chip communication to fetch the search window samples for all the reference frames (from one up to four in the case of the MVC prediction structures presented in Section 2.5.3). Without any optimization, the ME/DE requires the access of the entire search window (typically a 193x193 samples matrix) for each current block that is being encoded. It leads to impressive communication requirements for the ME/DE module, as it can be seen in Figure 2.13, where the MVC encoding modules (x-axis) are listed in terms of the required memory bandwidth (y-axis) (SHAFIQUE, ZATT, *et al.*, 2012) .

Considering the results of Figure 2.13, the ME/DE process is responsible for almost 70% of the MVC encoder memory bandwidth. The next most memory bandwidth consumer is the intra prediction, with 7%. The ME/DE is not present in the MVC decoders, so the memory accesses are more equally distributed among all modules. The module that most requires data from memory in the decoder side is the deblocking filter, representing 31% of the overall memory accesses. As conclusion from this discussion, *the challenges of saving the memory energy by reducing the off-chip memory communication are all mainly related to the ME/DE module.*

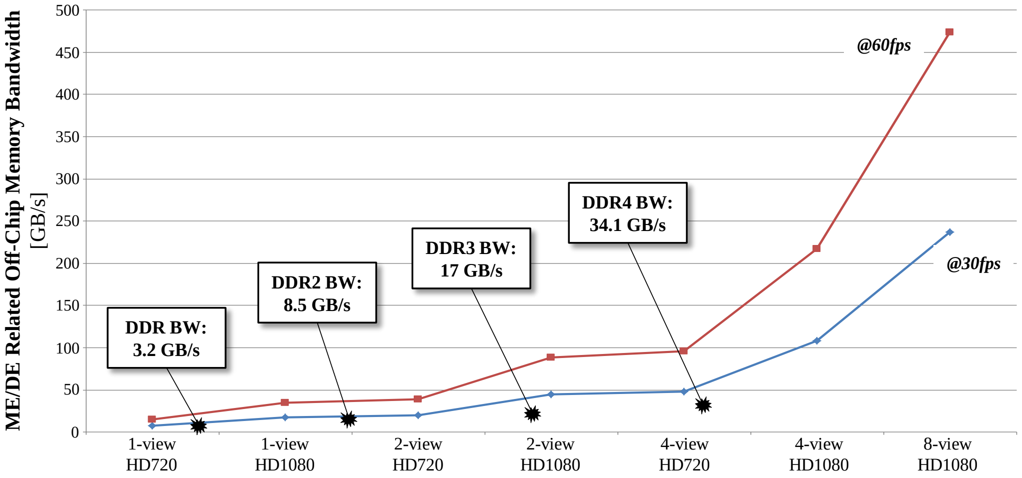


Figure 2.14: ME/DE off-chip memory bandwidth trend for several MVC scenarios.

The intensive number of memory access that is required by the ME/DE module, besides the energy consumption, restricts the MVC encoder implementation due to the DDR maximum data communication rates. Figure 2.14 presents a projection of memory requirements for several singleview and multiview encoding scenarios. The y-axis presents the off-chip memory bandwidth requirements only for the ME/DE module and the y-axis shows the evaluation for several scenarios. The parameters were: 193x193 search window size, IBP prediction structure (when allowed), and GOP size equals to eight.

The increasing curve of memory requirements is the reflex of the nowadays MVC encoder demand scaling due to (1) higher number of views (four up to eight views), (2) larger resolutions and (3) higher frame rates (30, 60 and even 120 frame per seconds). Even for a non-extreme scenario, “2-view HD1080” for instance, the memory access needed in the ME/DE processing surpasses the maximum data transfer allowed by the three already consolidated DDR off-chip memory technologies (DDR (MICRON, 2003), DDR2 (MICRON, 2004) and DDR3 (MICRON, 2006)). Besides, even the projection of maximum traffic allowed in the new DDR4 technology is not capable to deliver all the necessary data for the ME/DE on time. This way, *there is a strong need of memory bandwidth savings to become feasible the use of DDR off-chip memories to deal with the new MVC memory restrictions*.

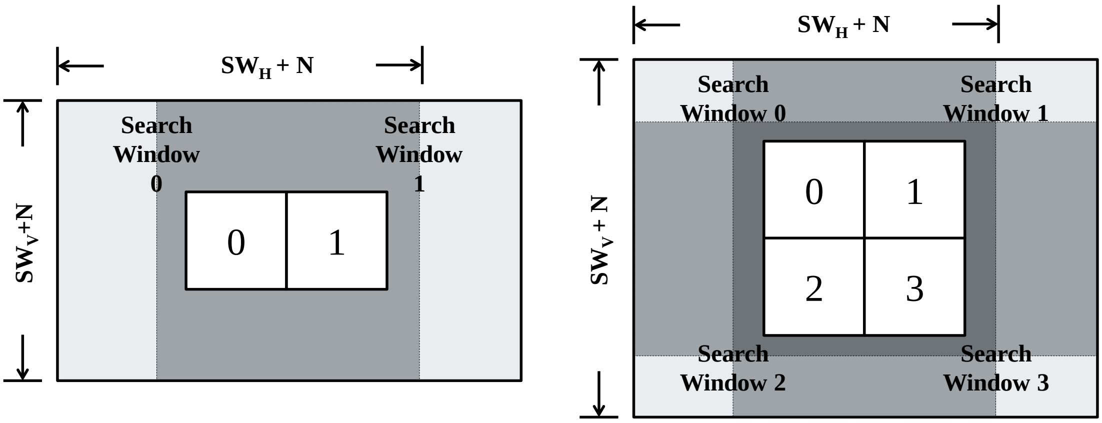
3 MEMORY AWARE TECHNIQUES FOR VIDEO CODING

Memory issues on digital video encoders have been a subject of research during several generations of non-MVC video encoders, like MPEG-2, H.263, and the latest H.264/AVC standard and its MVC extension. Many kinds of solutions were then proposed considering the state-of-the-art video coding standard at that time. Generally, as in this work, the inter prediction is the focus, more precisely the Motion Estimation.

In this work, for simplicity, these related works are divided in three groups: (1) data reuse strategies (local cache implementation), (2) reference frame compression approaches, and (3) hybrid energy-efficient approaches that implement different statistics based techniques. These works are presented in details as follows.

3.1 Data Reuse Strategies

In (TUAN, CHANG e JEN, 2002), the authors propose the Level A-D data reuse strategies, which exploit the data locality of the search window samples in both candidate block and search window levels. An incremental work (CHEN, HUANG, *et al.*, 2006) refines the initial proposed strategies and designs a more efficient scheme called Level C+. Figure 3.1 presents the Level C and Level C+ data reuse schemes, where the dark gray area corresponds to the overlapped samples between adjacent search windows.



(a) Level C

(b) Level C+

Figure 3.1: (a) Level C and (b) Level C+ data reuse schemes.

In Figure 3.1, N is the block size dimension (in the case of MB, $N=16$), SW_H and SW_V are the horizontal and vertical search window dimensions.

The work (GRELLERT, SAMPAIO, *et al.*, 2011) presents a multilevel approach based on the Level C+ strategy that reuses data between search windows of consecutive

MBs (SRAM memory banks) and between candidate blocks by exploiting the FS algorithm pixel-by-pixel based search (register buffer).

The works (CHEN, HUANG, *et al.*, 2005) and (TSUNG, DING, *et al.*, 2007) looks for the ME/DE memory problem from a different perspective. They changed the ME/DE processing order to reduce the number of times that one reference frame is fetched. Due to this out-of-order processing, undecided MBs need to have their partial results saved until the last ME/DE operation over that MB is performed. Both works targets the H.264/AVC and the work (TSUNG, DING, *et al.*, 2007) focused specifically on the MVC constraints.

Table 3.1 presents a comparison between the related work described in this section in terms of such aspects: (1) video coding standard target, (2) ME/DE perspective processing order, (3) data reuse level and (4) the partial results handling technique (if required). At the end, the proposed memory hierarchy specifications are described.

Table 3.1: Data reuse techniques comparison.

	Technique	**Target**	**ME/DE Perspective**	**Data-Reuse Level**	**Partial Results Handling?**
(TUAN, CHANG e JEN, 2002)	Level A-D	non-MVC	MB-centered	Candidate Block and SW	-
(CHEN, HUANG, *et al.*, 2006)	Level C+	non-MVC H.264/AVC	ME-centered	SW	-
(GRELLERT, SAMPAIO, *et al.*, 2011)	Multilevel C+	non-MVC H.264/AVC	MB-centered	Candidate Block and SW	-
(CHEN, HUANG, *et al.*, 2005)	SRMC	non-MVC H.264/AVC	Reference-centered	Reference Frame	No
(TSUNG, DING, *et al.*, 2007)	SWCS	MVC H.264/AVC	Reference-centered	Reference Frame	No
This Work	**RCDR**	**MVC H.264/AVC**	**Reference-centered**	**Reference Frame and SW**	**Yes (Partial Results Compressor)**

Among the data reuse schemes published in related works, only the work (TSUNG, DING, *et al.*, 2007) deals with the MVC restrictions for memory energy consumption. The memory hierarchy proposed in this work is a reference-centered approach, as it will be discussed further, while only two approaches use this perspective to reuse data, (CHEN, HUANG, *et al.*, 2005) and (TSUNG, DING, *et al.*, 2007). However, none of these two works properly consider the impact of partial results (resultant from the reference-centered approach) in the off-chip memory traffic and the on-chip memory size. The proposed memory hierarchy scheme is capable to reuse the reference samples in two levels (reference frame and search window level) considering the MVC prediction structures.

3.2 Reference Frame Compression Techniques

The work (SONG, ZHOU, *et al.*, 2010) proposed an adaptive reference frame compression scheme to reduce the external memory bandwidth consumption. A variable length coding is proposed to compress each processing unit in an efficiently way. Moreover, a compression mode decision was required to adaptively choose the best compression mode according to the image characteristic. The focus is the non-MVC H.264/AVC encoders and the evaluations were perfomed by using the JM 15.1 software.

The technique proposed in (GUPTE, AMRUTUR, *et al.*, 2011) is a lossy reference frame compression that aims to minimize the impact of quality while significantly reducing power bandwidth requirement. The strategy is a transformless approach that uses lossy reference for ME to reduce memory traffic, and lossless reference for the MC to avoid encoder/decoder mismatches (called error compensation). The error compensation calculates the error bound and stores it to compensate the ME errors in the MC. The work (GUPTE, AMRUTUR, *et al.*, 2011) focuses on the non-MVC H.264/AVC encoder and uses the JM as experiments platform. Besides, a hardware implementation for the reference frame compressor is presented and the DDR power savings are compared to the power overhead inserted by the compressor hardware.

In (WANG, CHANDA, *et al.*, 2012) the authors proposed a lossless encoder for image sequences based on JPEG-LS (lossless approach of JPEG static image compressor) defined for still images with temporal-extended prediction and context modeling. Besides, different compression algorithms for the reference frame compression, like JPEG, JPEG200 and near-lossless JPEG-LS, and their impacts on the memory requirement and the overall lossless compression ratio have been studied. There is not a video standard as target, so the experimental results were based on own C++ implementation of the proposed techniques.

The work (SILVEIRA, GRELLERT, *et al.*, 2012) proposed a lossless Reference Frame Context Adaptive Variable-Length Compressor (RFCAVLC) to compress the reference frame samples. The algorithm splits the reference frames in 4x4 blocks and compresses these blocks by using static Huffman tables (from one to four) in a context-adaptive way. The experiments do not focus on any specific standard and the results were taken from an authors' specific software implementation. Furthermore, FPGA hardware architecture was also designed to encode the data as soon as possible.

In (MA e SEGALL, 2011), it was proposed a hybrid frame buffer compression algorithm to reduce the memory bandwidth targeting energy reduced video coding. In their work, the authors decompose the full-resolution image into low resolution (LR) and high resolution (HR) components. The proposed technique was integrated to the non-MVC test model of the emerging High Efficiency Video Coding (HEVC) (JCT-VC, 2012).

Table 3.2 presents a comparison between the reference frame compression related works in terms of such aspects: (1) video coding standard target, (i(1)i) lossy or losses coding, (3) encoder/decoder mismatch due to the possible compression losses and, finally, (4) frame adaptive compression support.

Among the related works listed in Table 3.2, the reference frame compression technique proposed in this work is the first to deal with the MVC memory constraints. Along with the work (SONG, ZHOU, *et al.*, 2010), this work proposes an adaptive lossy/lossless algorithm that takes into consideration the image region characteristic. The encoder/decoder mismatch is avoided due to the error compensation technique with minimal penalties in the off-chip memory bandwidth savings.

Table 3.2: Reference Frames compression techniques comparison.

	Technique	Target	Lossy/ Lossess	Encoder/ Decoder Mismatch	Frame Adaptive?
(SONG, ZHOU, *et al.*, 2010)	Adaptive Partition Based	non-MVC H.264/AVC	Adaptive Lossy/ Lossless	In-loop	Yes
(GUPTE, AMRUTUR, *et al.*, 2011)	MMSQ-EC	non-MVC H.264/AVC	Lossy	Error Compensation	Yes
(WANG, CHANDA, *et al.*, 2012)	JPEG-LS	-	Lossless	-	No
(SILVEIRA, GRELLERT, *et al.*, 2012)	RFCAVLC	-	Lossless	-	No
(MA e SEGALL, 2011)	Hybrid FBC	HEVC	Lossy	Untreated	No
This Work	**Intra-Based Adaptive Quantization**	**MVC H.264/AVC**	**Adaptive Lossy/ Lossless**	**Error Compensation**	**Yes**

3.3 Energy-Aware Memory Techniques and Architectures for ME/DE

The work (ZATT, SHAFIQUE, *et al.*, 2011) presents a run-time adaptive energy-aware ME/DE architecture considering the MVC encoding. The architecture incorporates efficient memory access and data prefetching techniques for jointly reducing the off/on-chip memory energy consumption. Search maps from previous MBs processing are used to predict the ME/DE behavior for the current MB. Power gating is used to shut down parts of the on-chip memory depending on the prediction. The search window is not completely fetched to avoid unnecessary access of unused samples when fast search algorithms are not the targets.

Low power architecture for an on-chip multi-banked video memory for ME/DE targeting MVC encoders is proposed in (ZATT, SHAFIQUE, *et al.*, 2011). The memory organization was driven by an extensive analysis of memory-usage behavior for several videos. Besides, the architecture was implemented in standard cell technology in order to evaluate the overall energy. The memory restrictions for each MB are derived due to the inter-view and inter-frame correlations. When static disparity/motion MBs are processed, adaptive power gating works to shut down or reduce the energy supply of the less probable unused on-chip memory regions. The reference samples fetching is performed as the search engine requires, saving memory accesses for fast search algorithms, but causing irregular access pattern in the off-chip memory. The techniques were all integrated in a VLSI design and evaluated in terms of on-chip energy consumption.

In (SHAFIQUE, ZATT, *et al.*, 2012), the authors proposed an adaptive power management targeting the on-chip memory for the ME/DE in MVC. The energy-aware control checks in the so called 3D-neighborhood for the texture, motion and disparity properties to predict the behavior for the current MB ME/DE processing. The algorithm groups different MB of the current frame and predicts the highly-probable ME/DE search direction in order to power-gate idle memory groups. The VLSI design of this architecture is claimed to be energy efficient due the use of the proposed memory power management. Once more, the fast algorithms are considered and not the entire search window is fetched, causing irregular off-chip memory access patterns.

Table 3.3 presents the comparison between the energy-aware techniques proposed in related works in terms of: (1) target video coding scenario, (2) techniques to save on-chip memory energy, (3) techniques to reduce the off-chip memory energy, (4) ME/DE perspective processing order, and (5) the off-chip access pattern (regular or irregular). At the end, the proposed memory hierarchy aspects are described.

Table 3.3: Energy-aware techniques comparison.

	Target	**On-chip Energy Reduction Techniques**	**Off-chip Energy Reduction Techniques**	**ME/DE Perspective**	**Regular Off-chip Access?**
(ZATT, SHAFIQUE, *et al.*, 2011)	MVC H.264/AVC	Power-Gating	Search Map pre-fetching, Candidate Level Reuse	MB-centered	No
(ZATT, SHAFIQUE, *et al.*, 2011)	MVC H.264/AVC	Power-Gating (multiple sleeping states)	Candidate Level Reuse	MB-centered	No
(SHAFIQUE, ZATT, *et al.*, 2012)	MVC H.264/AVC	Power-Gating (multiple sleeping states), ME/DE direction prediction	Candidate Level Reuse, ME/DE direction prediction	MB-centered	No
This Work	**MVC H.264/AVC**	**Power-Gating (multiple sleeping states), RCDR scheme, Candidate Merging**	**RCDR scheme, Search Window Level Reuse (regular access pattern), Reference Frame Compressor**	**Reference-centered**	**Yes**

Among the related works on energy efficient solutions targeting the MVC encoding, this work is the unique of using the reference-centered ME/DE perspective, which is highly mandatory for the MVC prediction structures with lots of frame dependencies. Besides, the proposed strategy scans the off-chip memory in a regular way, allowing good energy efficiency even by dealing with more data bandwidth.

3.4 Reference-Centered Schedule Motivation

Figure 3.2 shows the number of accesses for one ME/DE search on one given reference frame. It is possible to note that some pixels in the reference frame are accessed more than 1.000 times. The scenario becomes more challenging for MVC because its prediction structure demands the search on multiple reference frames to achieve efficient compression, as already discussed on Section 2.7. Both prediction structures presented in the Figure 3.3 are simplified versions from those presented in Section 2.5.3. They were remembered for easier explanation.

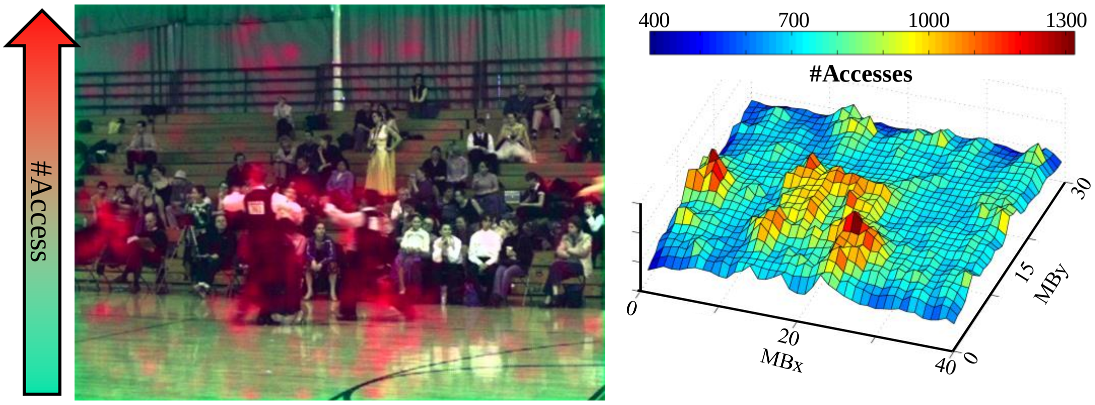
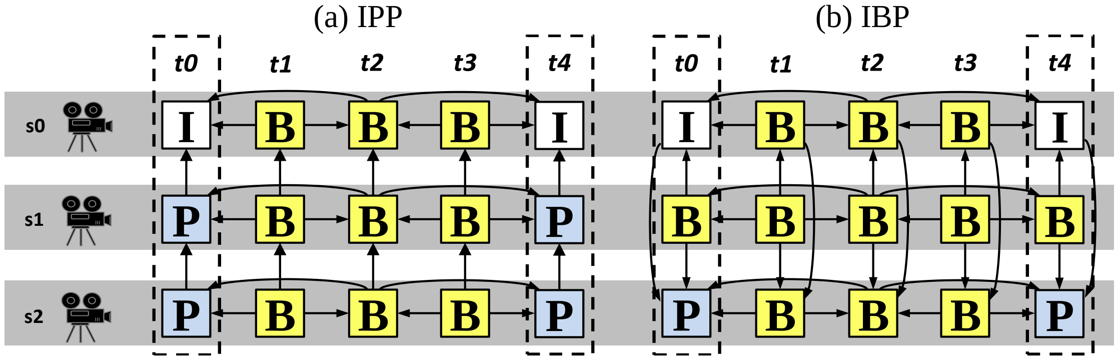


Figure 3.2: Memory Access Analysis


(a) IPP

(b) IBP

Figure 3.3: MVC Prediction Structure, MB-Centered Perspective.

Figure 3.3 presents the MVC hierarchical prediction structure at traditional current MB-centered perspective, where the origin of the arrows represent the current processed frame and the arrow head points to the requested reference frame. Examples for IPP and IBP view coding are provided. For simplicity, it will use 3 views and GOP equals to 4 in this explanation. However, for real application the GOP is typically higher and more views may be used (MERKLE, SMOLIC, *et al.*, 2007). Note that many frames are referenced multiple times to predict other frames. For instance, frame *s0t0* is referenced three and four times for IPP and IBP orders, respectively. It leads to a multiplication on the number of memory accesses (for MB-centered search) shown in Figure 3.2. Moreover, as MVC requires multiple reference frames, they must be simultaneously stored on-chip. For instance, frame *s1t1* requires four references to perform the complete ME/DE in the IBP case.

Naturally, reference data is not fetched on demand from off-chip memory every time it is required. As already presented, the data reuse techniques reduce the off-chip memory accesses by partially storing the data on-chip. In Figure 3.4, the design space corner cases are presented for the on-chip vs. off-chip tradeoff considering 8 views, GOP=8, IBP and search window equals to 193x193.

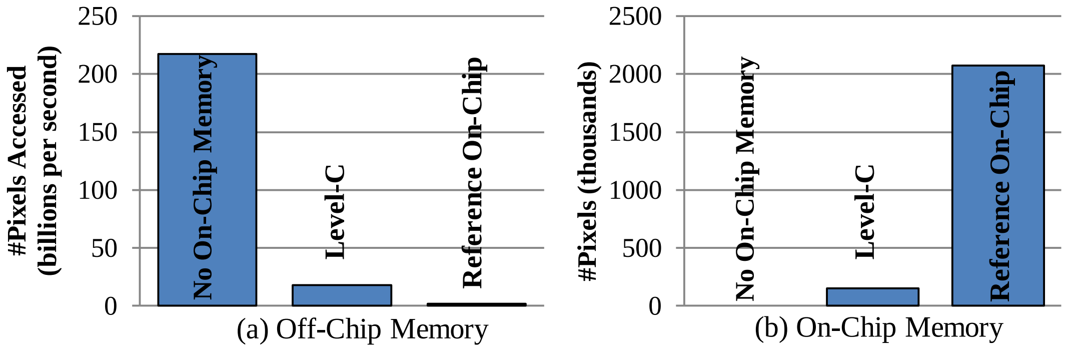


Figure 3.4: (a) Off-chip memory bandwidth vs. (b) on-chip memory size tradeoff.

In case no on-chip memory is employed, a huge off-chip memory bandwidth is required. In contrast, if the all reference frames are stored on-chip, the on-chip memory grows drastically. Level C (CHEN, HUANG, *et al.*, 2006) presents an intermediate and compromise in this tradeoff but large on-chip memory and numerous off-chip memory accesses are required due to MB-perspective limitation. To address this issue, the proposed memory hierarchy of this work employs a reference-centered data reuse scheme.

4 PROPOSED MEMORY HIERARCHY SYSTEM

This work designed a memory hierarchy to deal with the new challenges on energy-restrictions imposed by the MVC encoders. The goal is to jointly minimize the off-chip and the on-chip memory-related energy consumption to reduce the overall energy of the ME/DE on MVC. The off-chip memory traffic of the reference samples was reduced by the proposed Reference-Centered Data Reuse (RCDR) strategy, which locally reuses the accessed data of one reference frame as much as possible. The regular off-chip memory access pattern to fetch the search window samples was planned to be more energy-efficient than irregular approaches even when less data are fetched. The on-chip energy is minimized due to proposed low-power techniques on the on-chip video memory that reduces the supply voltage of the statistically less used regions. Besides, as multiple ME/DE are performed in the same search window, this work also proposed a candidate blocks merging to change the ME/DE evaluation order to reduce on-chip dynamic energy. Finally, a low-complexity reference frame compression algorithm that fits into the regular search window fashion of the memory hierarchy is proposed.

The proposed techniques were already published in high quality conferences: Design, Automation & Test in Europe (DATE), IEEE International Conference on Image Processing (ICIP), Design and Automation Conference (DAC) and Southern Programmable Logic Conference (SPL). The full text of these papers are presented in the Annex A.

All the static evaluations to determine the Huffman tables, the quantization levels and the thresholds definitions are described in the Appendix B of this Master Thesis.

This chapter describes the proposed memory hierarchy designed for the MVC memory restrictions. Initially, the system overview is presented by pointing the functionalities of each module. Then, each part is described in details: (a) Reference-Centered Data Reuse (RCDR), (b) Statistic-Based Partial Results Compressor, (c) On-chip Video Memory Organization, and (d) Intra-Based Reference Frame Compressor,

4.1 Memory Hierarchy System Overview

Figure 4.1 presents the architecture of the video memory hierarchy proposed in this work. The role of each module inside the Memory Hierarchy is explained as follows:

On-Chip Video Memory: SRAM memory that stores the search window samples and implements the Reference-Centered Data Reuse (RCDR) to reduce the off-chip memory bandwidth. It is composed by several banks according to the search window size and it is capable to store and to deliver one entire MB in parallel to feed the ME/DE operation parallelism. Sleep transistor were designed to implement the run-time adaptive power gating control to reduce the supply voltage of unused regions, thus saving leakage energy.

Search Control: It is responsible to generate the candidate blocks position in accordance to the TZ Search algorithm (XIU-LI, SHENG-KUI e CAN-HUI, 2010). These positions are sent to the Address Generation Unit to be translated into a burst of memory accesses. Besides, the Search Control also generates the current MBs positions for the search window that is currently stored on-chip and, after the best match is found, the Search Control sends the best motion/disparity vector and the best SAD to the Partial Results Compressor unit. The Candidate Merging technique is implemented inside this module.

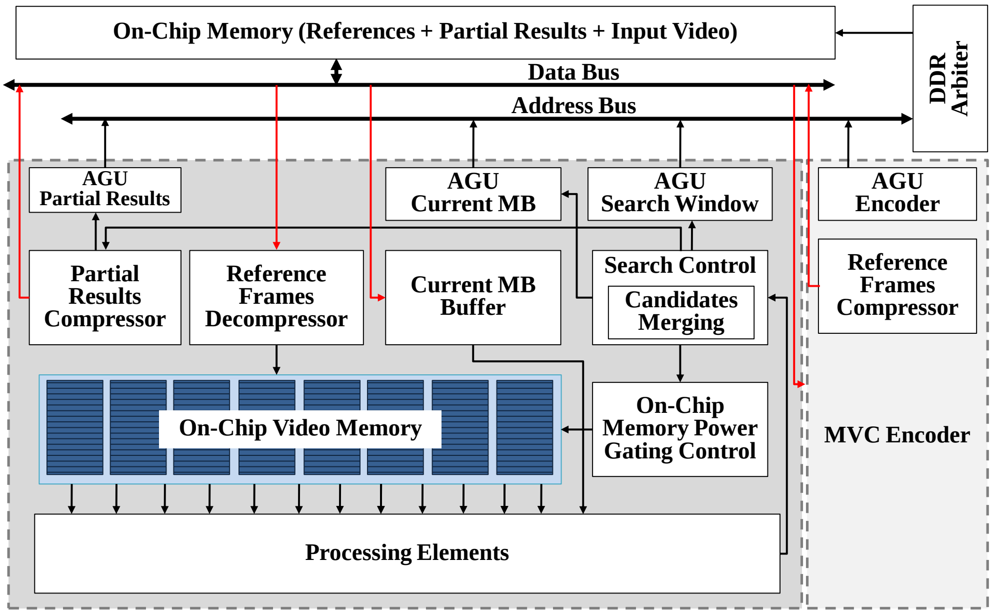


Figure 4.1: Overall Memory Hierarchy System.

Partial Results Compressor: RCDR imposes out-of-order ME/DE processing and partial results are generated: motion/disparity vectors and SAD costs. The Partial Results Compressor reduces the partial results bit representation to be efficiently stored in the off-chip memory until they can be discarded.

Address Generation Unit (AGU): Several Address Generation Unities (AGUs) were designed to handle the off-chip memory addressing to fetch or to save data: AGU Partial Results, AGU Current MB, AGU Search Window and AGU Encoder. They work in a synchronized way to schedule the AGUs memory access at different moments.

Current MB Buffer: Stores the current MBs for ME/DE that are dependent on the stored search window to be processed based on the RCDR order.

On-Chip Memory Power Gating Control: Run-time adaptive control that performs some statistics about the ME/DE on-chip access for the previously encoded frame. The goal is to predict the less probable accessed regions of the on-chip video memory to reduce their power supply voltages saving leakage energy.

Reference Frame Compressor/Decompressor: Besides all those techniques already discussed to save off-chip memory energy due to the reduced number of access, the Reference Frame Compressor/Decompressor modules aims to aggregate off-chip communication reduction by reducing the representation of the reference samples. The compression is performed at the final MB reconstruction loop before the reconstructed

MB is stored in the off-chip memory. On a similar way, the decompression runs after the search window fetching to recover the original samples.

Control and Data Flow: In order to control the ME/DE search, and, consequently, the memory access pattern, a Search Control is defined. Firstly, some requests are sent by the Search Control to fetch the search window samples from the off-chip memory. These requests are represented in the video positions format. Therefore, the AGU is used to translate the request into multiple physical memory positions. Then, the AGU sends the requests to DDR address bus, and the data is accessed through the data bus. Once the data is stored on-chip, a burst of candidate block positions is generated by the Search Control according to the TZ Search algorithm. These candidate positions are rearranged by the energy-aware candidates merging unit in order to reduce the on-chip memory line switching. An on-chip memory power gating control monitors the search statistics and power-gates the on-chip memory lines accordingly. The candidates are processed by an array of processing elements (the array is not inside the scope of this work). The best matching candidate and its SAD are forwarded to the Search Control. Due to the out-of-order processing inherent to RCDR, the temporary motion/disparity vectors and SAD values must be stored for further mode decision. These partial results are compressed using statistic-based non-uniform quantization and Huffman coding. As the partial results compressor employs variable-length coding (VLC), the partial results data is only sent to external memory once the local buffer is full. A specific AGU is implemented for partial results data.

In view of the possible memory contention created by multiple AGUs requesting access to the external memory, a fixed memory access scheduling presented in Figure 4.2 is also proposed.

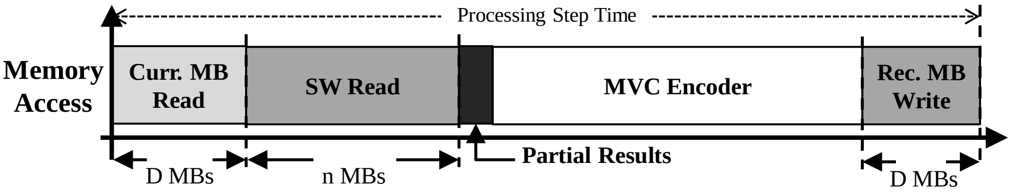


Figure 4.2: Off-chip Memory Access Scheduling

Initially, the current MBs are fetched in order to allow ME/DE processing start (a large fraction of the search window is already on-chip) followed by the missing search window column reading. In the following, if the partial results buffer is full, the partial results are sent to the external memory. Finally, the memory is available for other MVC modules. The reconstructed MB write step occurs in the end of the MVC encoding process. This work deals with this required bandwidth by compressing the reconstructed samples before they are stored in the off-chip memory (proposed reference frame compressor savings). Note that the number of current MBs (*D*) and the search window blocks (*n*) vary at frame level changing the duration of the schedule intervals. Still, the schedule is not affected.

The following sections will present in more details each one of the memory hierarchy modules and the proposed techniques.

4.2 Reference-Centered Data Reuse

The Reference-Centered Data Reuse (RCDR) is based on the inverted dependency logic between the reference frames and the current MB. In this approach, the reference search window is fetched from off-chip memory and those current MBs requiring that specific data are processed. In other words, the reference data "calls" the MBs to be processed, which is the opposite perspective when compared to the traditional scheduling, where a current MB "calls" several entire search windows to be fetched, named as Current-MB Centered Data Reuse (MBDR) in this work. In the scope of the RCDR, it is defined the term *dependent frames* for those frames "called" by a given reference frame. Figure 4.3 depicts the distinctions between search window-based MBDR and RCDR.

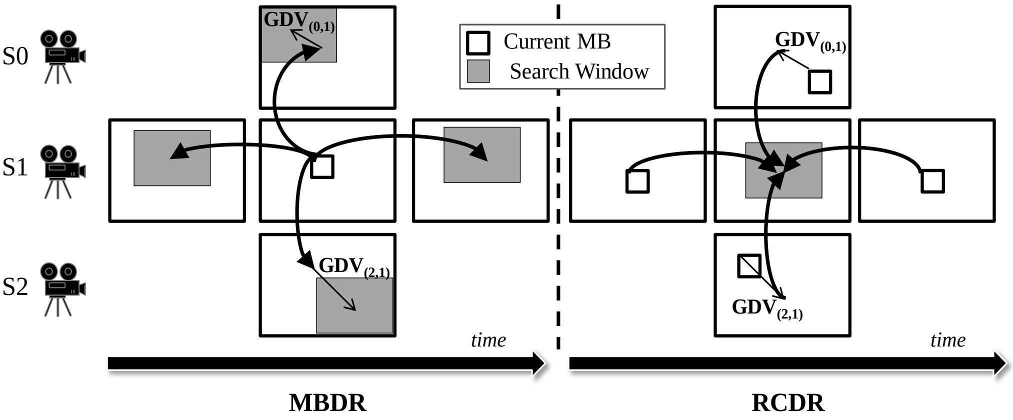


Figure 4.3: MBDR vs. RCDR Data Reuses for ME/DE.

Observe that in MBDR each current MB requests up to four search windows demanding 4x increased on-chip memory. Additionally, as those search windows will be required more times in the future (to encode other frames), external memory data retransmission is needed. In contrast, on-chip storage of a single search window is required for RCDR resulting in reduced on-chip memory. Moreover, in RCDR the search window is requested and read for external memory a single time. Indeed, current MBs belonging to dependent frames are accessed multiple times. However, this represents a small impact for both on/off-chip memories, since the search window dimension is generally very larger than the current MBs dimension. For the DE for example, the common test conditions specifies a minimum search window of [-96,+96), representing 169x more samples than the 16x16 fixed MB size. Additionally, the Global Disparity Vector (GDV) is taken into consideration to locate the current MBs positions, as shown in Figure 4.3.

In order to theoretically quantify the gains of the RDCR strategy against the MBDR, some mathematical analysis was performed. The main tradeoff here is to balance the off-chip memory access that must be as low as possible, along with the on-chip memory size that must be as small as possible. For simplification, it is considered only a squared search window with horizontal and vertical dimensions multiple of 16 samples. It means that the search window can be expressed as an integer number of MBs. For instance, the search window size [-96,+96) of a fixed-size 16x16 block can be expressed as function of the equivalent number of MBs, which is 169.

Equations (9) and (10) represent the on-chip (*OC*) memory size (in number of MBs) for MBDR and RCDR, respectively. In these definitions, *n* represent the horizontal (or vertical) search window dimension in number of MBs, *R* denotes the number of reference frames and *D* the number of dependent frames. Considering the MBDR, the quadratic term related to the search window size is then multiplied by the number of reference frames leading to an accentuated on-chip memory increase.

$$OC_{MBDR} = Rn^2 + 1 \tag{9}$$

$$OC_{RCDR} = n^2 + D \tag{10}$$

Equations (11) and (12) show the off-chip bandwidth (BW_{step}), in number of MBs, for each *search step* after the on-chip memory is completely full. The search step is defined as one MB processing for the MBDR scheme and one search window processing in case of RCDR. For each step MBDR must fetch from the off-chip memory R search window columns and one current MB. Meanwhile, the RCDR approach must read one search window column and D MBs. Note that the MBDR suffers with the search window increase. The bandwidth for a frame line (BW_{line}) is obtained applying Equations (13) and (14). BW_{line} is composed of two main components: (1) the initial fetch cost required to fill the on-chip video memory (first term); and (2) the *W-1* steps for reading the columns to complete the search window for the next MB. *W* is the horizontal frame dimension expressed as a number of MBs. It is important to notice that the R factor multiplies *n²* and Wn components in case of MBDR, leading to strong bandwidth increase with the number of reference frames and search window size. The bandwidth for one entire frame is obtained by multiplying the results from the Equations (13) an (14) by the vertical frame resolution in number of MBs (*H*) (Equations (15) and (16)).

$$BW_{step(MBDR)} = Rn + 1 \tag{11}$$

$$BW_{step(RCDR)} = n + D \tag{12}$$

$$BW_{line(MBDR)} = (Rn^2 + 1) + [(W - 1)(Rn + 1)] \tag{13}$$

$$BW_{line(RCDR)} = (n^2 + D) + [(W - 1)(n + D)] \tag{14}$$

$$BW_{frame(MBDR)} = BW_{line(MBDR)} \times H \tag{15}$$

$$BW_{frame(RCDR)} = BW_{line(RCDR)} \times H \tag{16}$$

These formulas are related to the calculation of the on-chip memory size and off-chip memory bandwidth considering the MBDR and RCDR itself. However, this work proposes compression techniques to reduce the representation of the partial results (RCDR penalties) and the reference frame samples. Although these terms are not listed in the Equations (11) to (16), they are important to consider for the overall bandwidth reduction calculation. Specially, the partial results will contribute only for the RCDR bandwidth, since the MBDR does not generate this overhead. On the other hand, the reference frames compression will improve the off-chip memory bandwidth savings for (i) the storage of the reconstructed MBs at the final of the MVC encoder loop and for (2) the search window read operation to fetch the reference data for ME/DE processing. These aspects are discussed in the next sections.

4.3 Statistic-Based Partial Results Compression

The partial results are composed of two very different data types, (1) motion/disparity vectors and (2) SAD values, which present distinct numerical range and statistical behavior. For this reason, we discuss them separately.

Figure 4.4 presented some statistical analysis performed in this work to understand the behavior of the motion/disparity vectors. The data set for these charts were extracted during the JM reference software encoding using the *flamenco2* test sequence (JVT TEAM, 2006). The Figure 4.4a demonstrates by using a 3D motion/disparity vector histogram that there is a sparse distribution of the *x* and *y* vector coordinates along the entire range. Although the motion vectors are mainly concentrated when *V=(0,0)*, multiple vectors are distributed in a wide range complicating the compression step. The 2D histogram emphasizes this idea by presenting the histogram only for the *y* coordinate distribution.

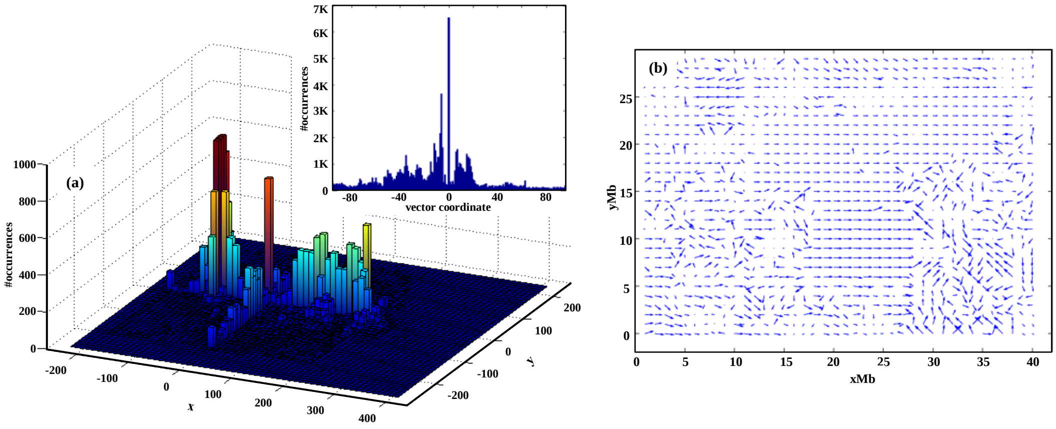


Figure 4.4: Motion/Disparity vectors statistical properties: (a) 2D/3D histogram and (b) vector field.

In order to concentrate these vectors in a reduced range, the similarities between the motion/disparity vectors of spatial neighbor MBs were exploited. This spatial correlation can be observed in the Figure 4.4b. Several different predictors were initially analyzed and the median spatial predictor (above and left MBs) provided the best results. This predictor is similar to the applied for the motion vector prediction defined in the H.264/AVC (RICHARDSON, 2003). In this approach, *differential vector* is generated as the difference between the original vector and the spatial predicted vector.

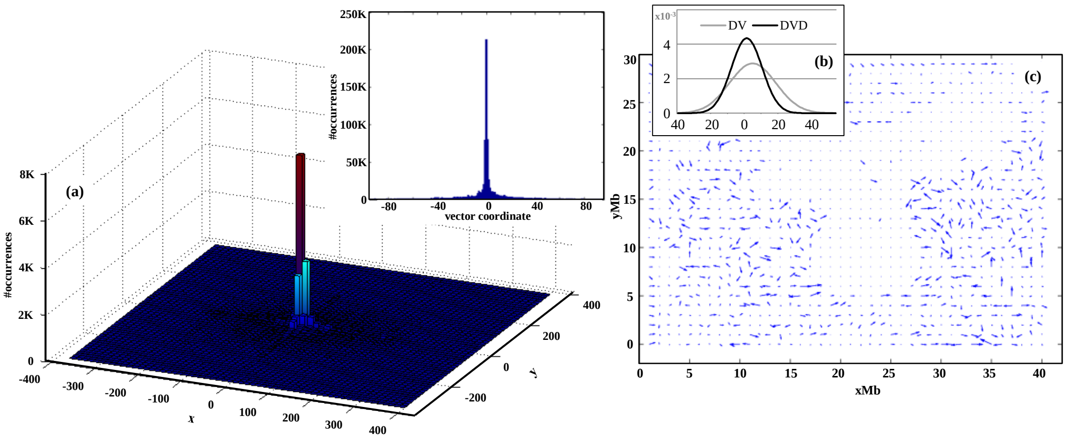


Figure 4.5: Differential Vectors statistics: (a) 2D/3D histogram and (b) differential vector field.

Figure 4.5a presents the 2D/3D histogram of the differential vectors distribution for both coordinates and specifically for one coordinate. The differential vectors are highly concentrated in the point *DV=(0,0)*, as shown in the PDF (Probability Density Function) in Figure 4.5b. It allows a good distribution to be compressed by a Huffman-based algorithm. In the spatial domain, the Figure 4.5c shows the differential vectors behavior, where the spatial correlation between neighbor vectors was exploited generating differential vector near to the zero vector.

Considering the differential vectors distribution, static Huffman tables were designed to reduce the bit representations (HUFFMAN, 1952). For 54 values that are inside the range $\pm(\mu + 2\sigma)$, where μ is the arithmetic average and σ is the standard deviation, which represents 95.8% of differential vectors occurrences, there are valid Huffman-entries to assign small codes to most probable vectors coordinates, in accordance with the statistical analysis. The differential vectors values out of this range are represented by a special Huffman value followed by the vector value in 8-bit binary representation. The methodology to generate the static Huffman tables is detailed on Section 5.1.

In Figure 4.6a the histogram and PDF of SAD values that must be stored for the ME/DE processing under the RCDR scheme are presented.

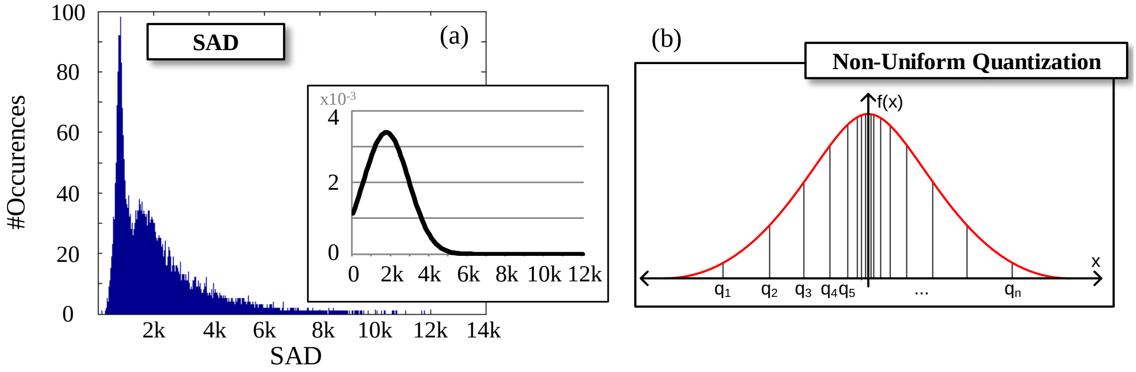


Figure 4.6: SAD Statistics for ME/DE: (a) histogram and PDF of SAD values and (b) non-uniform quantization approach.

As can be noted, SAD values are spread along a wide range of integer values range. Still, high concentration is observed around 1,000. For such kind of distribution, quantization is required for good compression gains. In order to reduce the impact of the quantization errors, we employ a non-uniform quantization designed for a Gaussian distribution in according to Lloyd algorithm and a Lloyd-Max refinement (MAX, 1960). The quantizer reduces the SAD representation range to 512 levels optimized for minimum mean squared error (MSE). The non-uniform quantization employs reduced quantization steps in high occurrence regions (close to the arithmetic average) and larger quantization steps in ranges with less occurrences (PDF tails), as represented in Figure 4.6b. After quantization, the quantized SADs are encoded by using a 189-entries Huffman table. SAD values out of range are encoded by a special Huffman value followed by the SAD value in 14 bits. After that, the partial results were concatenated in a 512-bits local buffer. Once this buffer is full, it is written to the external memory following by the schedule defined in Figure 4.2. The Partial Results AGU generates serial addresses in a memory region specific for partial results.

RCDR and partial results compression are focused on energy reducing for the off-chip communication. Besides, the reference frame compression acts in the same way by reducing the representation of the reference samples before they are stored in the external memory. The reference frame compression algorithm is detailed presented in the following sections.

4.4 On-chip Memory Organization

The proposed on-chip video memory aims at exploiting the regular off-chip memory access pattern to fetch the entire search window to save energy. Besides, as the TZ Search algorithm does not always require all the search window samples, this work also proposes a run-time adaptive power management to predicts the unused samples and to reduce the supply voltage of the corresponding on-chip memory positions.

The on-chip video memory is logically defined as a circular buffer organized in a 2D-array fashion to provide direct matching to video data. It is composed by *B* (where *B*=*n*) logical memory banks that rotates after each search step to avoid retransmission of overlapping search window samples. Figure 4.7a presents a simplified example with 3x3-MBs search window (each MB is represented by a distinct color) in time instant *T*. The rotation for time intent *T'=T+1* (after a *search step*) is presented in Figure 4.7b where the leftmost column of MBs (*MB(0,x)*) is dropped and a new column at the right is fetched (*MB(3,x)*). Columns MB(1,x) and MB(2,x) are reused. This organization, however, is not suitable for physical implementation once ME/DE requires MB parallel read.

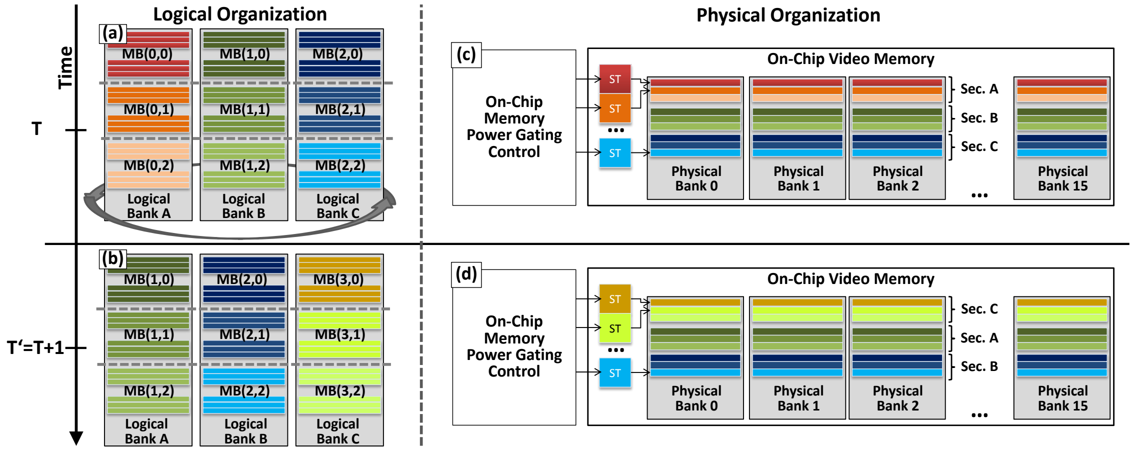


Figure 4.7: On-Chip Video Memory: (a)(b) logical and (c)(d) physical organization.

The physical organization of the proposed on-chip video memory is presented in Figure 4.7c and Figure 4.7d for time instants *T* and *T'=T+1*, respectively. It is composed of 16 parallel 128-bit wide SRAM banks to store 16 reference pixels per bank line. Each memory line stores and feed one complete MB in parallel. Each bank is further divide in sectors of *n* lines representing one search window column. The total number of lines is defined by the number of MBs in the search window (n^2). Note that differently from the logic organization, MBs columns are not shifted for every search step. To handle with it, the memory sectors are renamed accordingly, as depicted in Figure 4.7. Line-level power-gating is employed to support fine-grained power management, the power gating management is discussed in the next section.

4.4.1 Adaptive Run-Time Power Management

This work proposes a statistical power gating scheme that employs multiple SRAM sleep modes in order to reduce static energy consumption due leakage current. Differently from related work solutions (ZATT, SHAFIQUE, *et al.*, 2011), this strategy does not require image properties extraction and MB-level memory access prediction in order to provide an efficient solution. Four power states were implemented (SINGH, SYLVESTER, *et al.*, 2007): S_0=*OFF* (*V=0*), S_1=*Data Retentive* (*V=VDDx0.5*),

S_2=Data Retentive (*V=VDDx0.3*) and *S_3=Data Retentive* (*V=VDD*). Where each state has an associated wakeup energy cost (*$WE_{S0} > WE_{S1} > WE_{S2} > WE_{S3} = 0$*). For this reason, regions that are frequently accessed are mapped to S3, unused regions to S0, and other regions are mapped to S1-S2 according to run-time statistics.

```
1.  onChipPowerGating(n, v, CurrFrame, offStatMap)
2.  D {ME, DE} ← getNumberDependentFrames(v, CurrFrame);
3.  PowerMap_SW ← S0;                     // PowerMap initialization
4.  For all MB ∈ nxn                       // for all MBs in the nxn SW
5.        If (FirstFrame)       // if first frame
6.        Then      // use offline statistics
7.                  StatMap_SW = D {ME}*offStatMap_ME + D {DE}*offStatMap_DE;
8.        Else                  // use statistics from previous encoded frames
9.                  onStatMap ← getPrevFramesStat(v, CurrFrame);
10.                 StatMap_SW = D {ME}*offStatMap_ME + D{DE}*offStatMap_DE;
11.       EndIf
12.       PowerMap_SW ⎧ $S_1$  if   $\mu - 2\sigma \leq StatMap_{SW}(x,y) < \mu - 3\sigma$
                     ⎨ $S_2$  if   $\mu - \sigma \leq StatMap_{SW}(x,y) < \mu - 2\sigma$
                     ⎩ $S_3$  if   else
13. End For
14. PowerMap_SW ← physicalMemPos(PowerMap_SW);
15. For all MB ∈ CurrFrame               // for all MBs in the frame
16.       PowerGate(PowerMap_SW );
17.       currStatMap ← performSearch();
18. End For
19. StoreCurrMap(v, CurrFrame, currStatMap); return;
```

Figure 4.8: Pseudo-code of the on-chip video memory power gating.

Figure 4.8 presents our power-gating algorithm which is activated when one new frame processing starts. Firstly, the number of dependent frames is calculated (*line 2*) and the *$PowerMap_{SW}$* is reset. *$PowerMap_{SW}$* has one entry for each MB in the search window and if the used search window is smaller than the physical memory, all MBs exceeding the search window are fixed in *S_0*. For each MB in the search window (*line 4*), a weighted statistics map (*$StatMap_{SW}$*) is defines; offline statistics are used in the case this is the first processed frame (*line 7*), otherwise statistics from the previously encoded frames are used (*line 9*). The weighting factors depend on the number of ME/DE (*D{ME,DE}*) dependent frames. It is required due to distinct memory access behavior between ME and DE and can be noted in the plots of Figure 4.9.

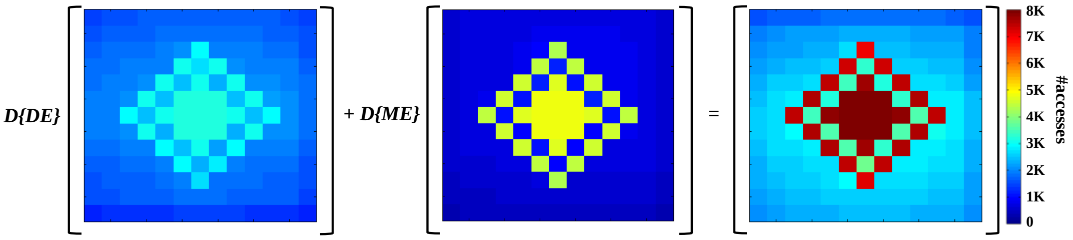


Figure 4.9: Statistical Maps of the TZ Search algorithm for each Motion/Disparity Estimation and the proposed weighted-calculation.

The *$StatMap_{SW}$* is then converted to *$PowerMap_{SW}$* by using statistically defined thresholds (*line 12*). The thresholds are calculated based on the memory access statistics (average and standard deviation) of each block within the search window. The *$PowerMap_{SW}$* is finally mapped to the actual physical memory positions (*line 14*). For each MB in the frame, the power gate signals are sent to the on-chip memory (*line 16*) and the ME/DE search is performed (*line 17*). At the end, statistics are updated for further frames processing (*line 19*).

4.4.2 Candidate Blocks Merging

Although static energy is becoming dominant in submicron on-chip memories, dynamic energy reduction significantly contributes to overall energy savings (RODRIGUEZ e JABOC, 2006). In order to avoid frequent on-chip memory line switching, we define an energy-aware candidate blocks merging strategy. As far as multiple dependent MBs are search simultaneously in the same search window, multiple search points are requested multiple times. The abstract example using TZ Search depicted in Figure 4.10 shows the access pattern for MB A (left) and MB B (right). Note that for the first search step (dark gray blocks) all candidates are the same. Additionally, some candidates in the second step (black blocks) are repeated. In the figure center, bright blocks represent candidate blocks accessed by both MBs and dark blocks are related to blocks accessed by a single current MB.

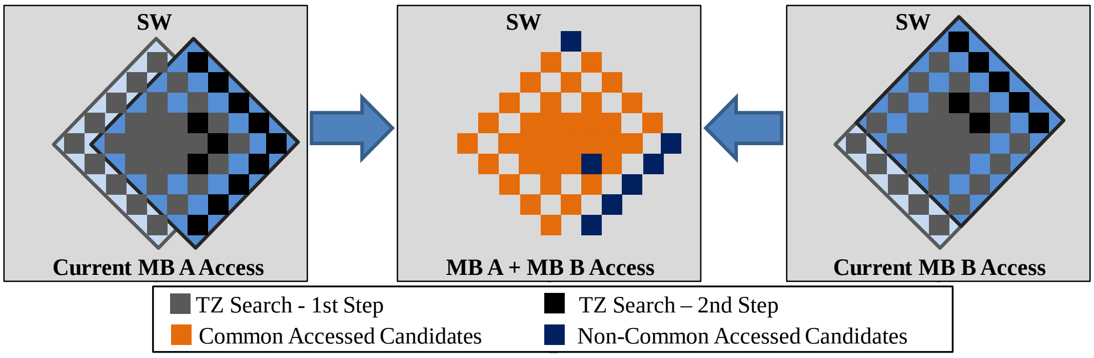


Figure 4.10: Example: Candidate Blocks Merging

The proposed candidate blocks merger receives all search points generated by the search control and rearranges them in order to process together repeated candidates and avoid unnecessary SRAM line switching (address line switching, bitline pre-charge, sense amplifiers switching, output buffer switching). Moreover, the proposed processing order follows the left-right and up-down fashion so the processing can start even before the rightmost column is updated for each search step.

4.5 Intra-Based Reference Frame Compression

Most of data that must be transmitted (read and write operations) from the off-chip memory for the ME/DE processing on MVC is related to the reference samples. This way, a further step is proposed in this work, which is the reference frame data compression to reduce the binary representation and, then, save off-chip memory energy. The compressor side acts to code the reconstructed MB (the final result of the MVC encoding loop) before it is written in the off-chip memory to be available for the future ME/DE. On the other side, the decompressor works to recover these reference samples to form the search window.

There are some requirements imposed by the proposed memory hierarchy system for this compression algorithm:

Low complexity algorithm: The compression algorithm must imply in as small overhead as possible in the MVC encoding loop. Besides, the overhead energy for the compression computation must be insignificant when compared to the energy savings in off-chip memory communication. *In this work, the reference frame compression exploits the spatial correlations by implementing a low-complexity intra-frame encoder* *that inherits the best intra mode of the MVC RDO mode decision, avoiding hard computation.*

Search window based: The algorithm must have a good fit with the proposed search window column scan order. *The intra-based reference frame compressor uses as reference the spatial neighboring samples. The proposed RCDR scans the reference frame in a regular order (column-by-column to complete the next search window), so the availability of the reference samples is always guaranteed.*

Adaptive due to frame characteristics: The technique must exploit the image characteristic in order to apply lossy/lossless compression with minimal drops in the MVC encoder rate-distortion final results. *This work considers information directly about the original frame (no data dependency with the MVC encoding loop) to capture the homogeneous/heterogeneous regions and to apply different non-linear quantization steps.* The goal here is to minimize the losses in the MVC coding efficiency.

MVC encoder/decoder mismatch: The inserted error in the MVC encoding loop must not be propagated for the output bitstream. In this case, the decoder will have different references from the encoder. *This work applies the compensation of the error, where the errors are stored and compensated during the Motion Compensation process* (GUPTE, AMRUTUR, *et al.*, 2011). In doing so, the errors will only affect the ME/DE performance (due to reference with such errors), but they will not imply in encoder/decoder mismatches.

Equations (17) and (18) models the off-chip bandwidth required for the RCDR along with the compression savings (CBW_{read} and CBW_{write}), where $BWframe_{(RCDR)}$ is the RCDR total memory access for one frame processing, as previously presented in Equation (16), α is the compression rate, and ε is related to the error compensation required for lossy compression to avoid encoder/decoder mismatches, W and H are the frame resolution in number of MBs.

$$CBW_{read} = \boldsymbol{\alpha}\, BW_{frame(RCDR)} + \boldsymbol{\varepsilon_{read}} \tag{17}$$

$$CBW_{write} = \boldsymbol{\alpha}\, (W \times H) + \boldsymbol{\varepsilon_{write}} \tag{18}$$

As the proposed compression algorithm deals with variable length codes, the compression parameter α in Equations (17) and (18) has a statistical behavior (average and standard deviation), depending on the video characteristics.

The proposed compression algorithm aims to extract all possible information during the MVC encoder process to avoid extra computation. In other words, the compression algorithm must *learn from the encoder knowledge what the best way to code the reconstructed MB is*.

This work exploits the spatial correlation between neighboring regions of the frame by performing a simplified intra prediction process over the reconstructed MB before they are stored in the off-chip memory. The raw reconstructed samples are spread along the 8-bit representation of luma and chroma representation. The goal is to use the intra prediction to generate residual information (difference between the predicted block and the input compression block). These residual samples tend to be concentrated in a small range of values, mainly for homogeneous regions where the intra prediction achieve the best results. Then, adaptive non-uniform quantization is applied to further reduce the representation range improving the compressor performance. Finally, static Huffman tables were designed to assign small codes to the most probable symbols.

The intra prediction processing for all modes was already discussed in the Section 2.6. as the background concepts of this Master Thesis were detailed with. For further algorithm-specific details, the H.264/AVC standard normalization text is referred (JVT TEAM, 2003). All static non-uniform quantization levels and Huffman tables were statistically extracted by using the *learning set* of multiview test sequences. These test conditions are summarized in Section 5.1.

The next sections will describe in details the proposed adaptive-quantization reference frame compression algorithm. The algorithm explanation is performed in a gradual way: first, a simple lossless approach without quantization is explained; then, a second algorithm is presented, which performs a static lossy compression, and finally, the proposed adaptive compression is presented by the composition of the first two approaches.

4.5.1 Lossless Compression

Figure 4.11 presents the compression algorithm for the basis lossless solution of the proposed reference frame compression technique. The lossless compressor is composed of four basic steps: (1) computation of the best intra mode that has been previously chosen during the MVC encoding process (*line 8*), (2) residue calculation by subtracting the reconstructed MB by the predicted block (*line 9*), (3) Huffman encoding to assign smaller codes to most probable values of residue and, finally (*line 10*), (4) the packing of this variable-length data along with the used intra modes into fixed-size packages to be sent to the off-chip memory (*lines 11-12*).

```
    compressLosslessReconMB(currentMB)
1.  /* currentMB: samples of the current MB
2.     predModes[16]: prediction  modes from the MVC encoder RDO
3.     neighborSamples: reference for Intra Prediction
4.  */
5.  predModes ← getIntraModes()
6.  codedMB ← Ø
7.  For each blk4x4 in currentMB loop
8.       pred4x4 ← intraPrediction(neighboring, predModes[blk4x4])
9.       resBlk4x4 ← pred4x4 – blk4x4
10.      codedBlk4x4 ← huffmanEncode( resBlk4x4, staticHuffTable)
11.      codedMb.append(predModes[blk4x4])
12.      codedMb.append(codedBlk4x4)
13. end loop
14. return codedMb
```

Figure 4.11: Lossless compression algorithm for reference samples.

In the decompressor side, which is responsible to recover the search window samples to be stored in the on-chip video memory, the inverse compressor steps are performed: first, (1) the data is unpacked and the residual data stream is separated from the prediction modes, so (2) the Huffman decodes the encoded residual information and recover the reconstructed MB, then (3) the intra prediction process generates the predicted block from the neighbor samples, and finally, (4) the reconstructed block is remounted due to the sum of the predicted block and the residue block.

It can be noted on Figure 4.11 (*line 7*) that the compression steps are performed for each 4x4 block of the reconstructed MB. To explain this decision of using only 4x4 block size, Figure 4.12 presents the residual information for the following intra prediction approaches: only I4MB modes (4x4 blocks only), only I16MB modes (16x16 blocks only), and at last, (c) the best RDO choice between I4MB and I16MB (both 16x16 and 4x4 are allowed).

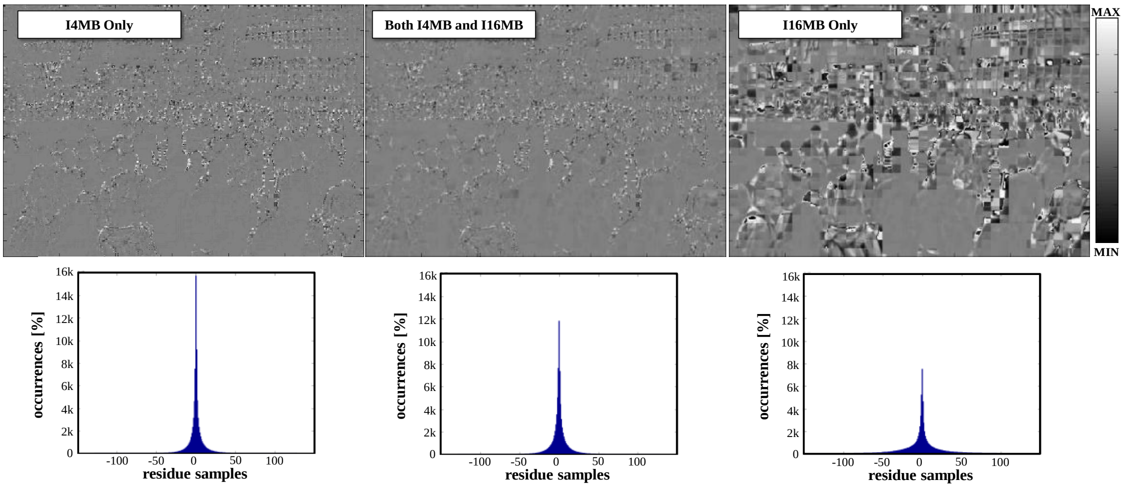


Figure 4.12: Residue plot examples from the *crowd* test sequence: I4MB only, I16MB only and both I4MB and I16MB modes.

The I16MB modes are the best RDO choice for homogenous image regions, since there are high spatial correlations and the bit overhead to represent the intra mode is minimal: only 2 bits is required for the entire MB samples (four modes are allowed for the I16MB). On the other hand, the MVC RDO decision choses the I4MB modes when lesser homogeneous are the target. Then, the residue values are lower than the I16MB (as seen in

Figure 4.12), but the overhead for the modes representation is higher: 16 modes with 4-bit length must be sent, since there are 9 possible prediction modes for I4MB. Meanwhile, the proposed reference frame compression path is a simplified approach to achieve high compression due to minimal possible computation. This way, the same rules that are adopted for the H.264/MVC RDO decision for the intra prediction process cannot be directly applied to this problem.

Experimental analysis pointed that the use of only I4MB provides the best compression, and consequently, the best off-chip memory communication reduction. It means that, even with a large overhead of sent sixteen 4-bit intra modes per MB, the compression of the residual blocks surpasses the other approaches, as can be noted in

Figure 4.12. For this reason, only I4MB prediction modes are supported in the reference frame compressor proposed in this Master Thesis.

The lossless compression does not imply drops in the MVC encoding process and it is the basic solution for the reference frames compression. Static Huffman table was designed to code the entire range of the residue representation, i. e., 256-entrie table is required. In order to improve the compression, the quantization step was inserted to reduce the representation range of the residue.

4.5.2 Quantization-Based Lossy Compression

For some video sequences, even the intra prediction residue is not as concentrated as necessary to provide as good variable-length coding efficiency as required for MVC encoding. In this sense, the lossy compression adds the quantization step inside the reference frame compression algorithm presented in the latest section. Then, statistics based non-uniform quantization was designed in this work to reduce the representation range and, this way, to improve the performance of the Huffman coding.

Figure 4.13 presents the lossy compression algorithm for the reference samples. The algorithm behavior is similar to that presented for the lossless approach (Figure 4.11). After the residual calculation, the quantization is applied to reduce the representation range of the samples (*line 11*). The quantized samples are Huffman-coded and packed along with the prediction mode. In order to be compliant with the H.264/MVC standard, the inserted errors must not be discarded and an extra step is necessary to perform the error compensation (*line 15*).

The quantization strength is defined by the *nLev* parameter, which corresponds to the number of non-uniform quantization levels that will be employed. For instance, *nLev=8* means that the [-128,+127] original representation range will be reduced to eight possible representations in the interval [-4,+3]. This leads to losses in the compression, since the original precision will not be recovered at all. This work evaluates three different values for *nLev={32,16,8}*. More levels were initially evaluated, but due to the concentrated distribution of the residue values, the results are practically the same when *nLev* is greater than 32.

```
    compressLossyReconMB(currentMB, nLev)
1.  /* currentMB: samples of the current MB
2.     predModes[16]: prediction  modes from the MVC encoder RDO
3.     neighborSamples: reference for Intra Prediction
4.     nLev: number of quantization levels (8, 16 or 32)
5.  */
6.  predModes ← getIntraModes()
7.  codedMB ← Ø
8.  For each blk4x4 in currentMB loop
9.      pred4x4 ← intraPrediction(neighboring, predModes[blk4x4])
10.     resBlk4x4 ← pred4x4 – blk4x4
11.     [quantedBlk4x4, errorsBlk4x4] ← quantize( resBlk4x4, nLev)
12.     codedBlk4x4 ← huffmanEncode( quantedBlk4x4, staticHuffTable)
13.     codedMb.append(predModes[blk4x4])
14.     codedMb.append(codedBlk4x4)
15.     compensateErrors(errorsBlk4x4)
16. end loop
17. return codedMb
```

Figure 4.13: Lossy compression algorithm for reference samples.

Figure 4.14 presents the residual plots of the lossless and lossy compression using 32, 16 and 8 quantization levels.

It can be noted that the residual values become near to the value 0 (center of the distribution) as strong is the quantization. Those heterogeneous regions that generate high residue values in the lossless residue plot (see Figure 4.14) were discretized to smaller values near the center of the distribution. The quantized values are promised to be very efficiently coded by the Huffman compressor, since the occurrences are highly centered on the average value and the smallest codes will be assigned to them.

Meanwhile, reduction of the representation range implies to errors for the decompressor side. The original samples before the compression process cannot be identically recovered and these errors will penalty the ME/DE processing, since the search will be performed into a modified reference. As results, this error will be propagate during the MVC encoding process and will affect the final rate-distortion coding efficiency. Still, the lossy compression must take it into account and should try to minimize these penalties. Figure 4.15 presents the histogram of the quantization errors for the three lossy scenarios: 8-levels, 16-levels and 32-levels. It can be note that the error is as less concentrated as stronger is the quantization. It means that 8-levels quantization provides the higher residual reduction (as observed in Figure 4.14) along with the higher losses in the MVC encoding due to the spread error distribution (presented in Figure 4.15).

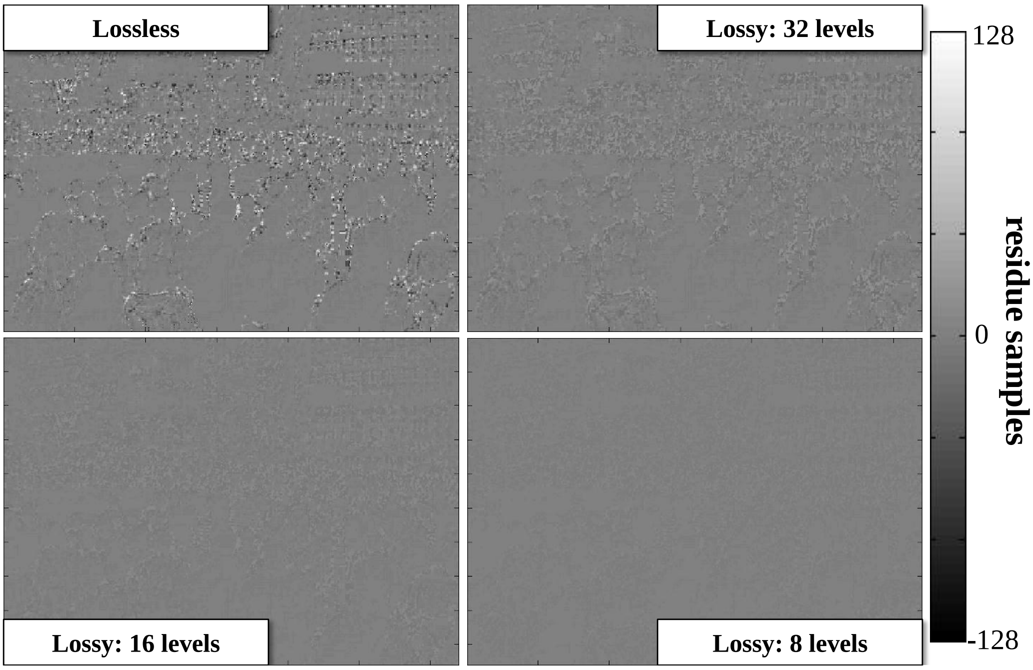


Figure 4.14: Residue plots for lossy compression for different quantization steps: lossless, 32 levels, 16 levels and 8 levels.

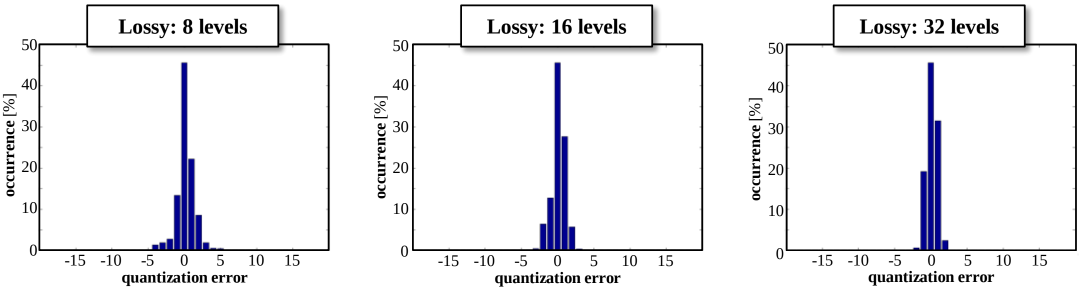


Figure 4.15: Histogram of the quantization errors for 8, 16 and 32 levels.

Figure 4.14 and Figure 4.15 summarize the main challenge for the reference frame compressors: *jointly minimize the required off-chip memory bandwidth on ME/DE (high compressing rates) and the MVC encoder rate-distortion drops*.

The lossless approach is one of the corner cases where there is no involved error (losses) in the ME/DE reference samples, but the compression is limited due to the not so centered distribution of the values. On the other side, the other corner case occurs when the strongest quantization is applied (in the case of this work, when *nLev=8*). In this scenario, the compression is the best possible due to the highly concentrated data distribution, but the quantization errors will provide the worst quality results.

In this sense, an adaptive approach which applies different quantization levels for different image regions depending on the MB characteristic is proposed. The goal is to predict the MB homogeneity to adaptively change the quantization strength. The idea is to reduce as much as possible the error keeping good compression rates. The next section will discuss the quantization-adaptive technique that is proposed in this Master Thesis.

4.5.3 Proposed Adaptive Quantization for Lossless/Lossy Compression

The adaptive approach exploits the statistics of the original MB samples to determine the intra prediction residue characteristic. Two types of information are derived from the MVC encoder: (1) the I4MB intra prediction modes for the sixteen 4x4 blocks and (2) the original 4x4 blocks variance. The variance calculation is performed over the original samples. In doing so, it does not dependent on any MVC encoding intermediate results and it can be calculated in parallel during the video coding process. Then, the residual blocks are classified depending on their homogeneity degree (given by the variance) and different quantization levels are applied to different classes of residues.

Figure 4.16 presents the proposed adaptive-quantization compression algorithm. The steps are quite similar to the fixed-quantization algorithm steps (see Figure 4.13). Thresholds were statistically defined to classify the residual information into four different homogeneity degrees, which will define the number of quantization levels that is going to be used to quantize the residue (*line 11*). Still, the non-linear quantization is used and the *nLev* parameter can assume the values [8, 16, 32, 256] for each 4x4 block. The value *nLev=256* means that no quantization is applied (lossless compression).

```
    compressAdaptiveLossyReconMB(currentMB)
1.  /* currentMB: samples of the current MB
2.     predModes[16]: prediction  modes from the MVC encoder RDO
3.     neighborSamples: reference for Intra Prediction
4.     nLev: number of quantization levels (8, 16 or 32)
5.  */
6.  predModes ← getIntraModes()
7.  codedMB ← Ø
8.  for each blk4x4 in currentMB loop
9.        pred4x4 ← intraPrediction(neighboring, predModes[blk4x4])
10.       resBlk4x4 ← pred4x4 – blk4x4
11.       nLev ← $\begin{cases} 8 & \textbf{if} \quad 0 \le \text{variance} < TH_0 \\ 16 & \textbf{if} \quad TH_0 \le \text{variance} < TH_1 \\ 32 & \textbf{if} \quad TH_1 \le \text{variance} < TH_2 \\ 256 & \textbf{if} \quad TH_2 \le \text{variance} \end{cases}$
12.       [quantedBlk4x4, errorsBlk4x4] ← quantize( resBlk4x4, nLev)
13.       codedBlk4x4 ← huffmanEncode( quantedBlk4x4, staticHuffTable[nLev])
14.       codedMb.append(nLev)
15.       codedMb.append(predModes[blk4x4])
16.       codedMb.append(codedBlk4x4)
17.       compensateErrors(errorsBlk4x4)
18. end loop
19. return codedMb
```

Figure 4.16: Adaptive-lossy compression algorithm for reference samples.

The thresholds *TH_0*, *TH_1* and *TH_2* were statistically defined to classify the residual blocks into four groups (*line 11*) depending on the variance value. Homogeneous blocks (low variance values) tend to have small residues since the intra prediction exploits this spatial neighboring correlation. In this case, the strongest quantization (*nLev=8*) is performed to concentrate even more the values to the center of the distribution. As the non-linear quantization performs small quantization steps to the values near the average, the error tends to be minimal. Heterogeneous blocks (high variance values) will generate residues values in a spread distribution. This way, more quantization levels are used in these blocks to reduce the errors.

The thresholds values were statistically determined by several experiments due to the *learning set* of videos defined in Section 5.1. The methodology used was: (1) fix several targets of maximum MSE allowed, then, for each 4x4 block, (2) adapt the quantization level until the quantization error fits in the target MSE, and finally, (3) the variance of the original block was inserted in the statistics for the chosen quantization level. The MSE value that provides the best compression along with the minimal errors was picked from the analysis and the thresholds were calculated due to the variance distribution for the four intervals as in the PDFs of Figure 4.17.

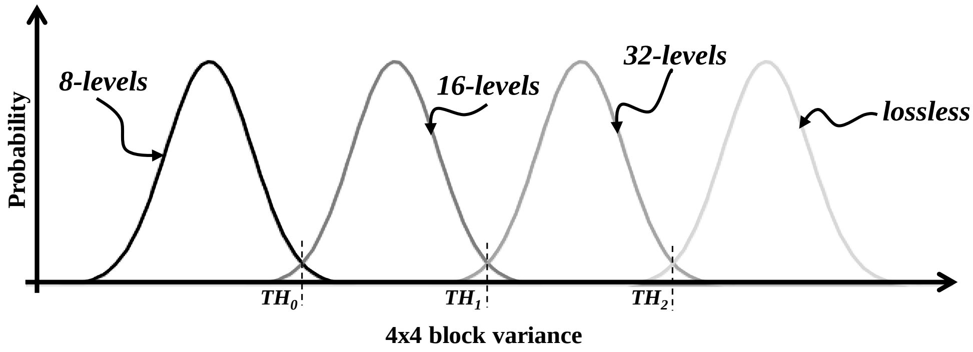


Figure 4.17: Thresholds definitions due to the statistical analysis of target MSE values.

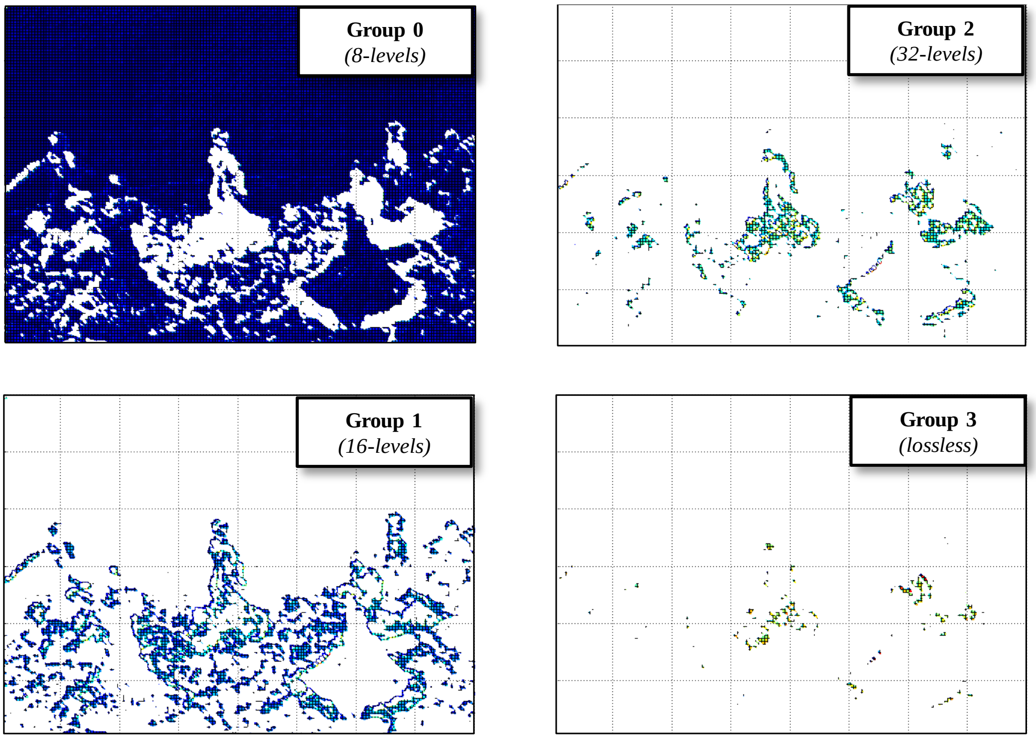


Figure 4.18: Classification of the residual data due to the variance (homogeneity) in the *flamenco2* test sequence.

Figure 4.19 presents the error reduction obtained by the adaptive-quantization compression algorithm in comparison to the static-quantization basis solution.

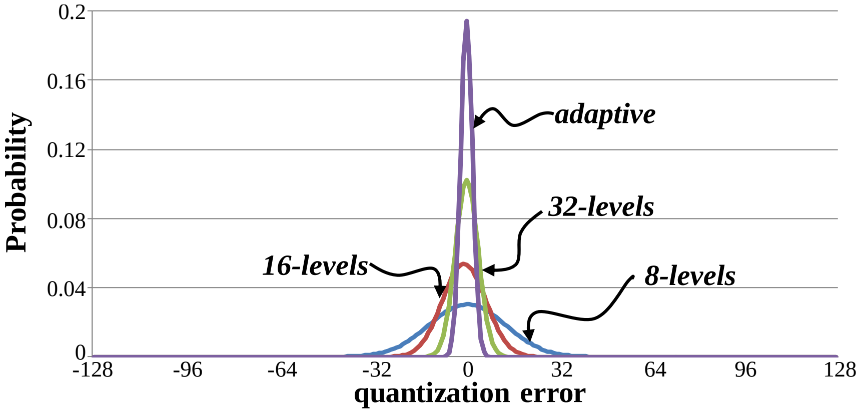


Figure 4.19: PDF for the errors considering *nLev=[8,16,32]* and the adaptive-quantization.

Figure 4.18 presents the classification of the 4x4 blocks considering their homogeneity by using a frame of the *flamenco2* test sequence as example. It is possible to note that the high homogeneous region of the video fit in the *Group 0*, which the 8-levels non-linear quantization will be applied. As higher is the variance of the target block as more quantization levels will be applied in the compression, in order to minimize the final error. Figure 4.18 also presents the other block classifications: *Group 1* (16-levels), *Group 2* (32-levels) and *Group 3* (lossless).

As can be noted, the errors are reduced due to dynamically change the quantization levels depending on the residue properties. For homogeneous blocks the residue tends to be small and concentrated to zero, so less quantization levels are applied to discretize the representation interval with minimal errors. On the other side, the heterogeneous regions are compressed with more quantization levels, maintaining the errors as similar as for the homogeneous blocks.

Chapter 5 will summarize the overall savings of both off/on-chip energy savings due to all described techniques inside the proposed memory hierarchy system.

5 RESULTS AND DISCUSSIONS

This chapter presents the experimental results to evaluate the proposed memory hierarchy itself and to compare it to related published works. The main goal of this work is to jointly reduce the on-chip and off-chip memory energy related to the ME/DE processing on MVC. The off-chip and on-chip energy savings are separately discussed to simplify the explanation.

Related to the on-chip memory, the proposed RCDR strategy along with the frame-level power gating control and the candidate merging technique provide the energy savings. The off-chip memory energy savings were provided again by the RCDR scheme, the partial results compression and the reference frames compression. All these savings are discussed in the following sections.

5.1 Experimental Tests Environment

The test environment for the MVC encoding was set to be compliant with the recommended common test conditions defined for the MVC research development provided by the standardization committees (JVT TEAM, 2006). The basis MVC encoder for the experiments was the reference software JMVC 8.5 (Joint Multiview Test Model) maintained by the JVT video coding group. The experiments included 3-views, 4-views or 8-views sequences considering the IBP and IPP prediction structures. Other common encoding settings were: CABAC, FRExt, QP={22,27,32,37}, GOP_{size}=8, and TZ Search algorithm.

The input video sequences were divided into two subsets. The first group, called *learning set*, is the subset of sequences used to perform all statistic evaluations in this work. For instance, the learning set was used as input benchmark to define the static Huffman tables, non-uniform quantization levels and thresholds for adaptive-quantization, off-line statistic map and thresholds for the power gating control. The multiview video sequences of this subset are: *crowd* and *race1* (640x480-VGA), *lovebird1* and *kendo* (1024x768-XGA), *poznan_street* and *dancer* (1920x1088-HD1080).

To effectively evaluate the performance of the proposed techniques, a second subset of video sequences were defined, called *evaluation set*. The goal was to generate the off-line parameters by using the learning set and apply them to the MVC encoding of the evaluation set of sequences. This set is composed of the sequences: *ballroom*, *flamenco2*, *exit* and *vassar* (640x480-VGA); *lovebird2* (720x576-SD); *newspaper* and *balloons* (1024x768-XGA); and *GT_Fly*, *poznan_carpark* and *poznan_hall2* (1920x1088-HD1080).

All the two subsets of test sequences are presented in Appendix A. Besides, objective indexes were calculated to measure three important characteristics of multiview videos: spatial index (SI), temporal index (TI) and disparity index (DI).

The energy savings results were obtained due to a customized simulation by considering datasheet nominal values of real DDR and SRAM memories (SINGH, SYLVESTER, *et al.*, 2007) (MICRON, 2007). The same methodology was applied to the techniques of related works to perform a fair comparison.

5.2 Off-chip Memory Energy Savings

The off-chip memory energy was reduced due to RCDR strategy that was implemented in the memory hierarchy system. The energy savings become from the reduced number of memory access and from the regular pattern that the off-chip memory is scanned. Reducing the number of off-chip memory accesses, the number of DDR *READ* operations will proportionally decrease. The regular pattern of the memory scanning provides more DDR *READ* operations in *bursts*, reducing the number of page activations (ACT). These two energy saving aspects are discussed as follows.

5.2.1 Memory Access Savings

Figure 5.1 presents the off-chip memory access reduction of the proposed RCDR scheme in comparison with the MBDR traditional ME/DE order using Level-C (CHEN, HUANG, *et al.*, 2006). The x-axis corresponds to the search window size, z-axis varies the video resolution and the y-axis presents the RCDR off-chip bandwidth savings over the MBDR scheme. Four scenarios were considerate: (a) 4-views IPP, (b) 4-views IBP, (c) 8-views IPP and (d) 8-views IBP.

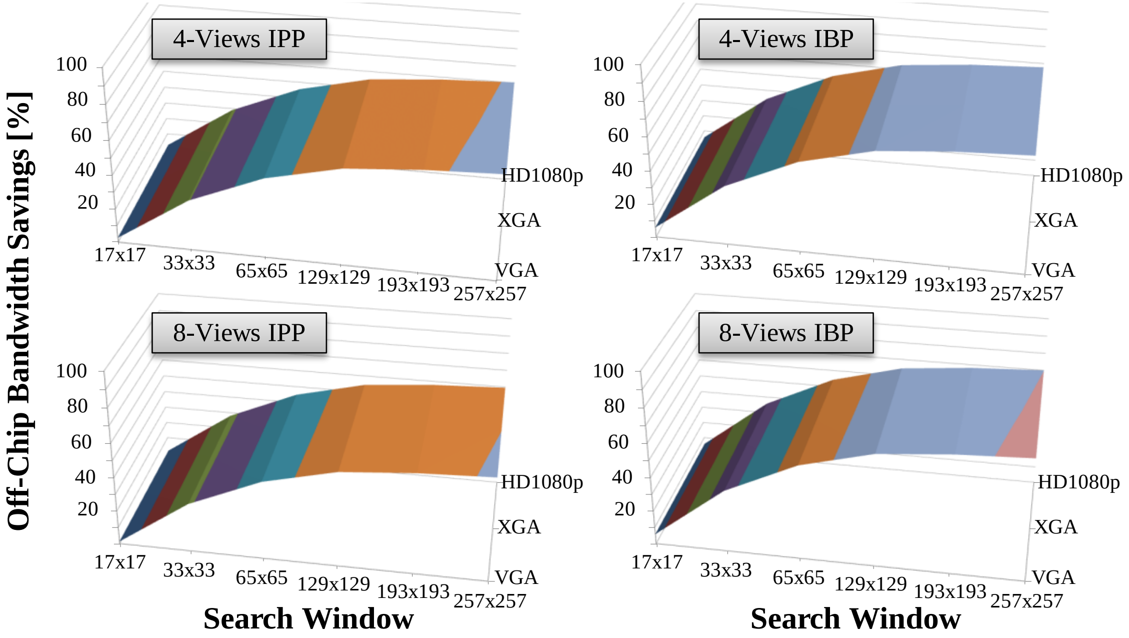


Figure 5.1: Off-chip memory bandwidth savings (related to the MBDR approach).

The proposed RCDR scheme presents off-chip memory bandwidth savings for all search window sizes and for all frame resolutions considered in this analysis. Considering the IPP prediction structure, the RCDR achieve gains from 2.7% to 60.3% for 4-views videos and savings from 1.3% to 59.5% in the case of 8-views sequences.

When the target prediction structure is the IBP, the savings vary from 5.8% to 67.8% for 4-views videos, and from 5.9% to 69.9% when 8-views sequences are processed.

Observe that the energy savings scale well with the increases in number of views and search window. Additionally, the proposed RCDR does not suffer with frame resolution increase. Higher savings happen in case of IBP due more intense search window reuse, i.e., each search window is used by an increased number of current MBs. As both approaches are based on the entire search window fetching, the memory access pattern regularity is about the same. Then, the energy savings are directly related to the off-chip memory bandwidth reductions presented in Figure 5.1.

The results in Figure 5.1 included the average savings related to the partial results compression, which deals with the out-of-order processing overhead (motion/disparity vectors and SAD costs). Figure 5.2 presents the off-chip memory savings due only to the Partial Results Compressor. The x-axis corresponds to several video sequences taken from the evaluation set and the y-axis gives the off-chip memory accesses savings corresponding to the partial results handling.

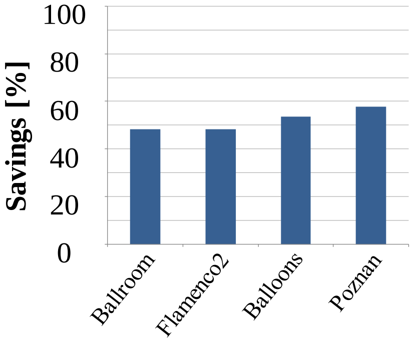


Figure 5.2: Off-chip memory bandwidth savings due to the Partial Results Compressor.

On the average case, the proposed compression algorithm is able to save 52% of the off-chip communication caused by the partial results transmission. Non-regular sequences with difference temporal/disparity behaviors, like *ballroom*, presented the worst case of the results (48% of reduction). On the other side, more regular videos provide more spatial correlation between neighboring motion/disparity vectors and SAD costs. So, the best case is achieved for the *poznan_carpark* video with 57% of savings.

5.2.2 Energy Savings

The off-chip DDR memory energy consumption considered in this work is based on the power components presented in the Micron® technical notes (MICRON, 2002). Basically, these technical notes derive the DDR energy consumption from (a) the READ/WRITE activity, (b) the page activations (ACT) needed to change the DDR page at each time instant, (c) the I/O DDR chip pads and (d) the standby overhead to maintain the memory on. Inside the same DDR page, the READ/WRITE operations can be executed in bursts, avoiding page activations and minimizing the energy consumption. As the proposed memory hierarchy is responsible to reduce the DDR read operations, only the read-related energy components are analyzed. The energy evaluations use the voltage and current parameters for the MT46H64M16LF memory by Micron Semiconductors© (MICRON, 2007). It is a 1 Giga-bit Low-Power DDR (LPDDR) memory.

Figure 5.3 presents the DDR energy savings due to the RCDR strategy when compared to Level C data reuse scheme (MBDR based). This energy savings already counts the extra overhead to transmit the compressed partial results (extra read cycles and page activations, on the average case). The x-axis represents the search window size, the z-axis varies the frame resolution, and the z-axis presents the off-chip energy savings itself.

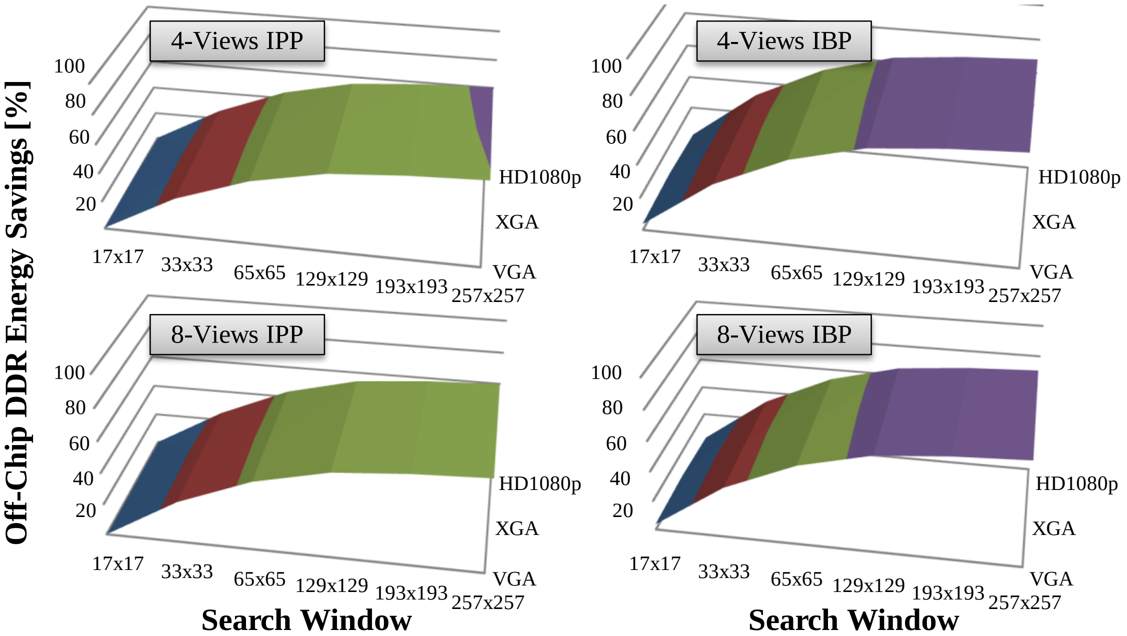


Figure 5.3: Off-chip DDR energy savings due to RCDR strategy (over MBDR).

The same trend from the off-chip bandwidth reduction (Figure 5.1) is reflected in the energy reduction. As already discussed, the search window scan fashion implies in regular memory accesses pattern for both RCDR and MBDR (with Level C). The reference MBs burst reads are well exploited by both approaches. This way, the energy savings are directly related to the saving of read cycles from the DDR memory. This deeper energy analysis is presented in the two analyses from Figure 5.4. The scenario for this analysis was: 4-views IBP (Figure 5.4a) and search window 193x193 (Figure 5.4b),

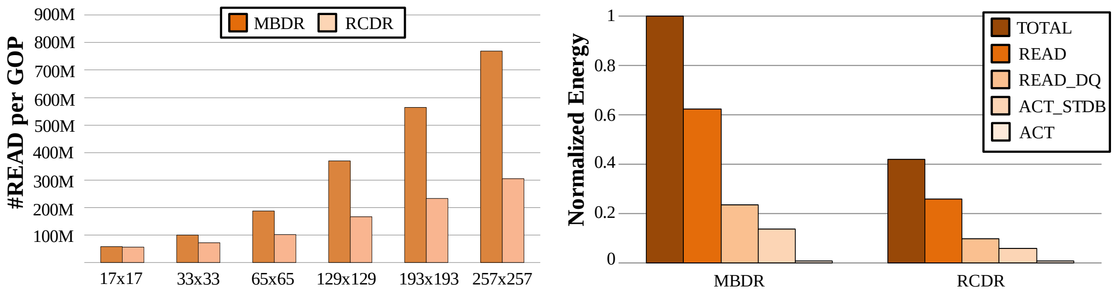


Figure 5.4: (a) Number of DDR Read cycles (#READs) and (b) DDR energy consumption profiling (RCDR vs. Level C).

Figure 5.4a presents the growing number of READ cycles per GOP (y-axis) due to the search window size increasing (x-axis). It can be noted the two distinct growing trends of MBDR and RCDR that directly reflects the energy savings presented in Figure 5.3. In the worst case, when the search window size is 257x257, the MBDR requires

2.5x more read operations than the proposed RCDR. It is directly reflected in the overall DDR energy consumption.

Figure 5.4b presented a decomposition of the DDR normalized energy consumption (y-axis) related to the read operations for both MBDR and RCDR strategies (x-axis). P_{READ} is the DDR internal energy to perform one read operation, P_{READ_DQ} is the energy to drive the data out to the DDR chip, P_{ACT_STDB} is the standby component to maintain the DDR in the active state (able to execute read commands), and P_{ACT} is related to the energy of activation the DDR pages. It can be note that the activation energy is not representative related to the energy spent for the effective read operations (less than 2%), since both approaches use a regular fashion and burst reads are easily allowed. The main energy savings, as already indicated, are provided by the reduction of the reading cycles, which is massively exploited by the reference frame level data reuse of the proposed RCDR scheme.

The regular DDR memory read pattern allows for the RCDR (search window based strategy) lots of read bursts operations inside the same physical page on the DDR. Related works, like (ZATT, SHAFIQUE, *et al.*, 2011), claim that the reduced memory bandwidth by not fetching the entire search window, when heuristic algorithms are the focus, will proportionally reflect in off-chip energy savings. However, these approaches read the DDR in irregular way, since the reference blocks are fetched on demand in according with the search algorithm. This fact is analyzed in Figure 5.5, where the energy components are listed and normalized to the total (ZATT, SHAFIQUE, *et al.*, 2011) energy (y-axis). The x-axis presents the total energy and the subcomponents (P_{TOTAL}, P_{READ}, P_{READ_DQ}, P_{STDB} and P_{ACT}) for the proposed RCDR and the search window formation technique (ZATT, SHAFIQUE, *et al.*, 2011). The scenario for this evaluation was: HD1080 4-views IBP and 193x193, TZ Search algorithm limited within 193x193 search window size.

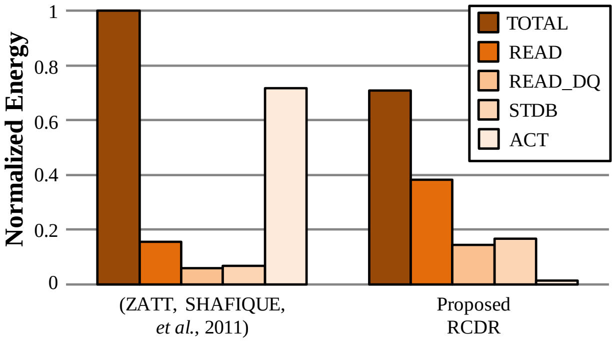


Figure 5.5: DDR energy consumption in comparison to (ZATT, SHAFIQUE, *et al.*, 2011).

It is possible to note that the proposed RCDR strategy is capable to reduce the DDR energy consumption in 30%. The effective read energy of the RCDR is 2.4x higher than (ZATT, SHAFIQUE, *et al.*, 2011), since only the required reference samples are fetched from the off-chip memory. However, the irregular fashion of the memory access of (ZATT, SHAFIQUE, *et al.*, 2011) leads with high energy consumption due to the required page activations and, this way, the DDR burst read is not well exploited. In terms of savings, the P_{ACT} is 98% reduced by the RCDR approach. As can be noted in Figure 5.5, the page activations is dominant in the overall energy consumed by (ZATT, SHAFIQUE, *et al.*, 2011). This way, even the RCDR requiring more read operations, the overall DDR energy consumption is reduced when compared to the irregular search based related work.

5.3 On-chip Memory Energy Savings

Besides the off-chip memory energy, this work also proposed techniques to reduce the on-chip energy consumption by: (1) reducing the on-chip memory size due to the RCDR technique, (2) power gating unused on-chip memory sectors to reduce static energy and (3) merging the candidate blocks to avoid SRAM line switching reducing the dynamic on-chip energy.

First, Figure 5.6 presents the on-chip memory size trend for RCDR in comparison to MBDR approach (using Level C). The x-axis varies the search window dimension and the z-axis shows the trend for three video resolutions.

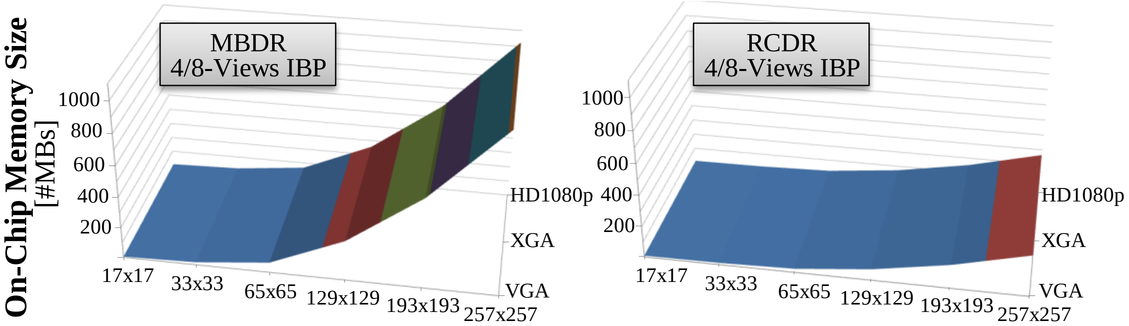


Figure 5.6: On-chip size increasing for RCDR and MBDR.

The proposed RCDR memory hierarchy presents the smallest on-chip memory size for all evaluated cases, except for the 17x17 where the current MB size is significant when compared to the search window size, which is not useful in practice. The best case is when the largest search window is used (256x256), where the reduction of 65% is observed for the IPP structure and 74% of savings for IBP prediction.

Compared to the Level C (MBDR based) approach, the proposed on-chip memory size is significantly reduced because there is no need to simultaneously store multiple search windows on chip. Note that the RCDR on-chip memory grows smoothly with the search window increase. Moreover, compared to the search window storage, the cost for storing current MB (that may reach 9 MBs for 8-views IBP) is negligible. This cost is amortized as the search window increases. The reduced on-chip memory size directly leads to less static energy consumption.

Besides the reduced on-chip memory compared to the Level C approach, the proposed memory hierarchy is also as energy-efficient as solutions that do not store all search window on-chip and, this way, implement smaller on-chip memory to cache the reference samples. This result is achieved due to (1) the proposed frame-level power gating control, to deal with the static energy, and (2) the candidate merging technique, to reduce the on-chip dynamic energy.

Figure 5.7 presents the comparison of the static energy (leakage) savings of the proposed on-chip video memory itself and with the power gating control savings along with related works: Level C, Level C+ (CHEN, HUANG, *et al.*, 2006) and (ZATT, SHAFIQUE, *et al.*, 2011). The x-axis presents the evaluation for different test sequences and the y-axis is the static on-chip energy normalized to the Level C+ consumption. The scenario for this analysis is: 4-views and IBP prediction structure.

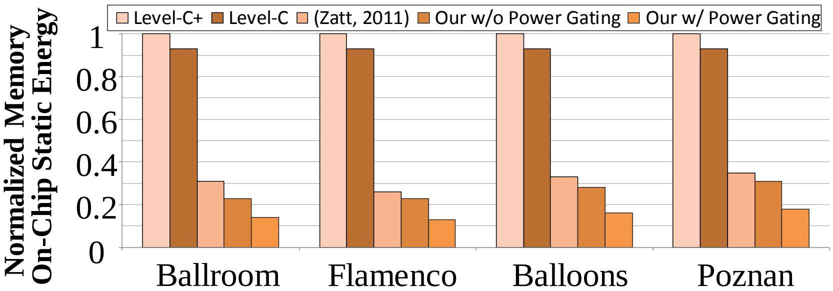


Figure 5.7: On-chip static energy savings (leakage).

As already expected, the Level C and Level C+, as they are based on the MBDR ME/DE traditional search order, they implements larger on-chip memory (as explained in the analysis of Figure 5.7) and consumes the highest static energy among the compared strategies. The work (ZATT, SHAFIQUE, *et al.*, 2011) implements a predictive approach that statistically defines the search window on-chip memory and it is also based in the MBDR scheduling. The proposed solutions, with or without the power gating control, is capable to reduce the on-chip static energy when compared to the related strategies for all tested sequences in the analysis. Compared to the Level C+, the worst case in the comparison, the proposed RCDR on-chip video memory consumes 77% less static energy without employing any power gating technique. In the power gating is on, further 88% of reduction is reached outperforming (ZATT, SHAFIQUE, *et al.*, 2011) in 61%.

The Candidate Block Merging approach acts to reduce the dynamic energy by reducing unnecessary on-chip memory accesses to fetch the same candidate blocks for current MBs that are searching in the same search window. Figure 5.8 summarizes the on-chip dynamic energy reduction (address line switching, bitline pre-charge, sense amplifiers switching, output buffer switching) provided by the proposed technique. The scenario for this evaluation was: 4-views IBP.

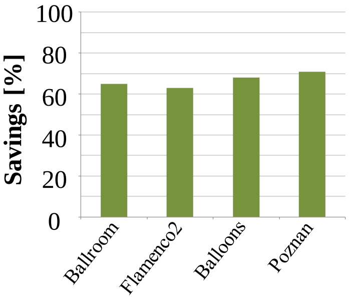


Figure 5.8: On-chip dynamic energy savings due to candidate merging approach.

On the average, the proposed technique achieved a dynamic energy reduction of 65%, as can be noted in Figure 5.8. At the best of the author’s knowledge, this is the first application specific technique to address the dynamic on-chip memory energy consumption for the ME/DE processing.

5.4 Reference Frame Compressor

This work also proposed a reference frame compression algorithm to reduce the representation of the reconstructed MBs (used as references for the next frames ME/DE) before they are stored on the off-chip memory. The decompression is performed when the search window samples are fetched, so there are off-chip bandwidth savings on both read/write operations from DDR. Indeed, the search window fetching is much more costly than the reconstructed MB storage.

As presented in Section 4.5, the proposed reference frame compression is an adaptive-quantization approach, where different number of quantization levels is applied to different image regions, in order to minimize the losses. As the intra-frame spatial correlation is exploited, the residual information differs from homogenous to heterogeneous MBs. In doing so, the variance of the original block is taken as homogeneity index and, in according to some statistical thresholds, the number of quantization levels are decided among *nLev={8, 16, 32, 256}*, where *nLev=256* represents no quantization (lossless compression).

Figure 5.9 and Figure 5.10 present the evaluation of the proposed adaptive compression algorithm in comparison with simple static solutions, where either the quantization is always the same for all 4x4 blocks (*8-levels*, *16-levels* and *32-levels* cases) or no quantization is performed (*lossless* case). The metrics to measure the efficiency of each solution are: (1) the compression rates, to indicate the potential of off-chip memory bandwidth reducing over the ME/DE operation, and (2) the quality difference between the reconstructed frames with and without the proposed compression losses (using the PSNR). The measurement (2) is not as the same as the final MVC encoding quality drops (more variables must be considered). However, this is a simple way to evaluate the adaptive solution in comparison with the static approaches.

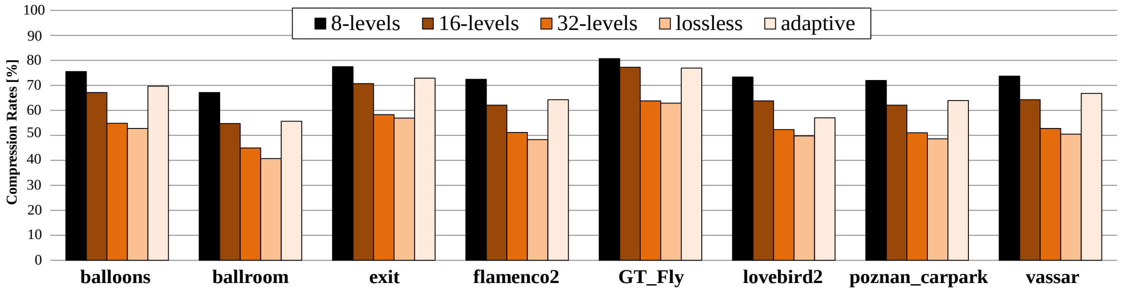


Figure 5.9: Off-chip bandwidth reduction of ME/DE reference samples due to the reference frame compression.

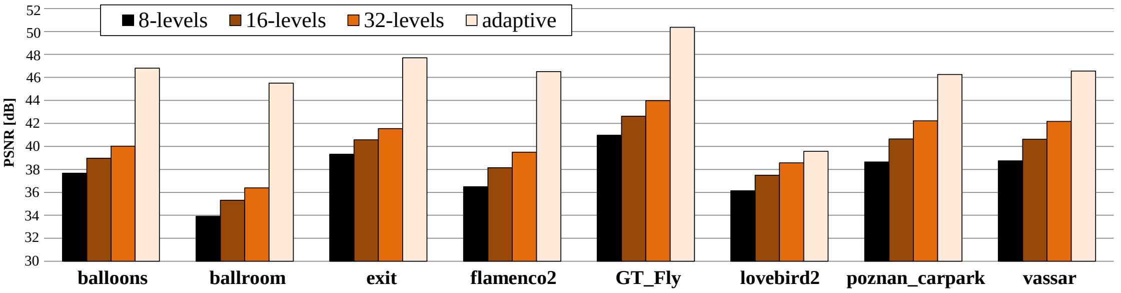


Figure 5.10: PSNR of ME/DE reference samples due to the reference frame compression.

Considering the compression results, the best result is achieved by the static 8-levels solution as expected, with savings of 74% on average. The adaptive solution achieves as good results as the 8-levels: 66% of reduction on average. Then, the 16-levels case saves 65%, 32-levels 53%, and the lossless approach is capable to reduce 51% of the representation on average. As can be noted by the average results and from the bars of

Figure 5.9, the adaptive solution can achieve compression rates as competitive as the static 8-levels quantization case.

The efficiency of the adaptive solution can be proved by taking this competitive compression results and comparing to the PSNR analysis (Figure 5.10), where the adaptive quantization solution is able to minimize the error by increasing the quality in 8.4 dB on average, when compared to the 8-levels static approach. The gains are 6.8 dB over the 16-levels and 5.6 dB over 32-levels. Besides, as the PSNR has a logarithm behavior, the absolute results of the adaptive-quantization indicates that this solution is able to cause negligible losses to the MVC encoder loop, as it will be discussed further.

The reference frame compression technique can be used along with the proposed RCDR-based memory hierarchy in order to provide even more off-chip memory bandwidth reduction. Figure 5.11 present the comparison between the MBDR (using Level C), proposed RCDR and RCDR with reference frame compression in terms of off-chip memory communication. The x-axis contains the communication components (ME/DE total, search window fetch, reconstructed MB written and error compensation) and y-axis present the normalized off-chip traffic to the MBDR total. The MVC encoder scenario is: search window size 193x193, 8-views IBP.

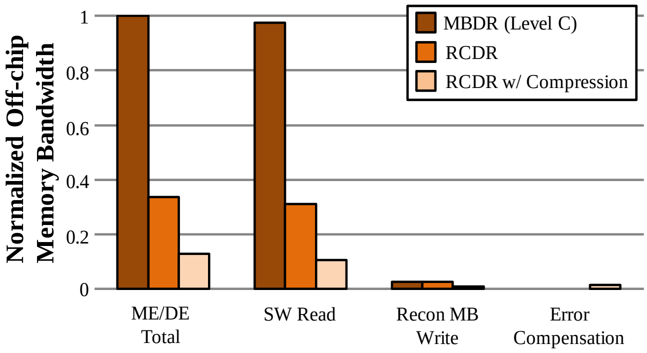


Figure 5.11: Off-chip memory bandwidth savings due to the proposed reference frame compression.

It can be note that the search window fetching is responsible for 97%, 92% and 83% of the total off-chip memory bandwidth, respectively for MBDR, RCDR and RCDR with compression. This ratio varies in according to the search window size, becoming less impacting as smaller is the search window. However, the typical dimension to catch the disparity correlation is about 193x193 (MERKLE, SMOLIC, *et al.*, 2007). Compared to the Level C, the RCDR with the reference frame compression is able to achieve savings of 83% on the off-chip memory communication, improving in 2.6x the RCDR reduction results. The error compensation bandwidth is not representative when compared to the ME/DE search window fetching, responsible for less than 10% of the total off-chip memory communication.

The insertion of losses in the MVC encoder loop generates a propagated error as far an anchor I-Frame erases the prediction dependencies. This way, inside the GOP, all disparity and temporal correlations are affected by the first error, since the last dependent frame of the last processed view depends on the first processed frame of the GOP. In doing so, this reference frame compression error must be as small as possible, to deal with this hard penalty.

Table 5.1 presents the comparison of the proposed reference frame compression algorithm when compared with state-of-the-art approaches: (MA e SEGALL, 2011) (GUPTE, AMRUTUR, *et al.*, 2011) (SONG, ZHOU, *et al.*, 2010) (SILVEIRA, GRELLERT, *et al.*, 2012). The proposed compression algorithm was integrated with the JMVC 8.5 reference software. As already discussed, the encoder/decoder mismatches is avoided by the error compensation technique. The Bjontegaard Delta rates (BDPSNR and BDBR) (BJONTEGAARD, 2001) was adopted to evaluate the MVC encoding efficiency. The analysis considers 4 test sequences (*ballroom*, *flamenco2*, *objects* and *poznan_street2*), *QP={22,27,32,37}*, IBP prediction structure and 3-views processing.

Table 5.1: Comparison with Reference Frame Compression Related Works

Sequence	**Target**	**Off-chip Memory Bandwidth Savings**	**BD-PSNR**	**BD-BR**
Proposed (lossless)	**MVC**	**51.3**	**0 dB *(lossless)***	**0% *(lossless)***
Proposed (adaptive-quantization)	**MVC**	**69.5**	**-0.01 dB**	**+0.18%**
(MA e SEGALL, 2011)	non-MVC	21% 31%	N.I.	+0.38% - +21%
(GUPTE, AMRUTUR, *et al.*, 2011)	non-MVC	17% to 24%	-0.01 dB	+0.74%
(SILVEIRA, GRELLERT, *et al.*, 2012)	non-MVC	24%	0 dB *(lossless)*	0% *(lossless)*
(SONG, ZHOU, *et al.*, 2010)	non-MVC	25% to 50%	-0.044 dB - -0.56 dB	+1.36% - +3.92%

This work is the only one that focuses on the MVC prediction structures constrains among all related works in the comparison. Due to the high dependency of the frames (temporal and disparity correlations) in the MVC prediction structures, the errors are propagated not only through the temporal neighboring frames due to the ME dropped references, but also in the disparity domain in the DE operations. This way, the MVC issues on the error propagation is harder than the non-MVC scenario. However, this work succeed in achieving high off-chip memory communication savings, which is required for the impressive reference samples transmission on MVC, along with negligible penalties in the MVC encoder efficiency.

Among the lossy approaches, this work presents the highest off-chip memory bandwidth savings, 69.5%. When compared to the only lossless solution published in (SILVEIRA, GRELLERT, *et al.*, 2012), this work surpasses the off-chip memory traffic reduction (51.3% against 24%). The penalties on the rate-distortion performance are the smallest one among the related works: 0.01dB of PSNR quality drop and 0.18% of bistream overhead. It proves that the proposed adaptive-approach is successful to minimize the error by dynamically choose the quantization strength in according to the video characteristics. The fully lossless algorithm itself is also a good option, since the error compensation overhead is not required and competitive savings on the off-chip memory communication is achieved (the best among the lossless solutions).

6 CONCLUSIONS AND FUTURE WORKS

This work presented a memory hierarchy for the Motion Estimation (ME) and Disparity Estimation (DE) modules to jointly minimize the on-chip and off-chip memory energy targeting the Multiview Video Coding (MVC) memory requirements. The MVC encoding process requires even more data transmission to the off-chip memory due to more sophisticated prediction structures to handle with the inter-view correlations between views captured for different cameras. The most energy consuming modules of the MVC encoder are the ME and DE (ME/DE), representing up to 98% of the total energy. Moreover, considering the ME/DE energy only, 90% is spent for the off-chip memory transmission to fetch the search window samples (45%) along with the on-chip memory that keeps the search window locally store to feed the ME/DE search (45%).

Several techniques were proposed to minimize the off-chip and on-chip memory energies in a jointly way. First, the Reference Centered Data Reuse (RCDR) was proposed to exploit the data reuse in the reference frame level. The idea was: once the search window is fetched, all dependent MBs that must be processed by the ME/DE using the already fetched search window are computed together. This leads to an out-of-order processing and generates partial results (motion/disparity vectors and SAD costs). In order to reduce the partial results transmission, it was proposed a statistics based partial results compression to reduce the off-chip memory bandwidth to save these overhead from RCDR. Besides, an energy-aware on-chip video memory was also implemented to handle with the RCDR strategy. This internal memory is capable to locally store the entire search window on chip, following the RCDR processing order. Besides, the fetching of the entire search window provides a regular off-chip memory access pattern. So, the proposed RCDR strategy takes advantage from this to perform lots of burst reads to fetch the data from the off-chip DDR memory. It is well-known that the DDR energy consumption is lower when burst reads are performed (less energy on page activations). As fast search algorithms do not require all search window samples, it was designed a power management power-gating control to reduce the power supply of unused regions and save static energy. Besides, the dynamic on-chip energy was reduced by using the candidate merging technique, which changes the candidate blocks evaluation order among different MBs searches. Finally, it was proposed a reference frame compression technique that fits into the proposed memory hierarchy requirements to improve even more the off-chip memory communication. It implements a low complexity intra based compression, which benefits from the MVC encoder knowledge to avoid computation and to implement an adaptive lossy/lossless approach. Depending on the image characteristic, the reconstructed blocks are classified to different groups, which lead to a particular quantization strength.

The experimental results proved that the proposed system accomplished the goal of jointly minimizing the energy consumption related to the off-chip and on-chip memories targeting the ME/DE on MVC. The RCDR strategy provides off-chip energy savings for all evaluated MVC scenarios (maximum of 68%), when compared to the state-of-the-art MB-perspective data reuse scheme Level C. The savings on off-chip memory bandwidth became from the reference-perspective of the RCDR. Besides, the savings growth together with the search window size and the number of reference frames that are used (MVC leads to lots of frame dependencies and large search windows). The partial results compression is able to reduce the off-chip memory traffic due to the transmission of the RCDR out-of-order processing penalties in 52%. The proposed RCDR strategy also achieved the best results even when compared to not search window based approaches, due to the regular off-chip memory access pattern allowed by the RCDR (column-by-column of the search window are read in a regular fashion). Evaluations using a LPDDR memory showed the energy reduction of 30% for the target MVC scenario. Besides, the off-chip memory energy is also saved due to the reference frame compression technique. When integrated to the RCDR memory hierarchy, the compression algorithm is capable to improve in 2.6x of the off-chip memory communication savings due to insignificant losses in the MVC encoding efficiency (rete-distortion tradeoff). Compared to the Level C MB-perspective search, the RCDR with reference frame compression leads with 83% of memory accesses reduction. The on-chip memory energy is saved also for several techniques. First, the RCDR scheme reduces the on-chip memory size up to 74% compared to the MB-perspective Level C, since the reference-perspective allows the storage of one search window at time. Besides, the on-chip video memory also implements a frame-level adaptive power gating management that achieves the static energy savings of 82% (best results among the related works). The dynamic energy is treated due to the candidate merging technique, achieving savings of more than 65%.

As future works, it is planned to evaluate the compressor computation overhead energies in order to measure the effective gains on the off-chip energy saving. Besides, the improving of the compressor algorithms by trying to catch temporal and disparity correlations is also planned. As a new course of study, the next steps of researching will take the memory energy related consumption for the 3D extension of the emerging video compression standard, the High Efficiency Video Coding (HEVC), targeting *manycores* implementations.

REFERENCES

BJONTEGAARD, G. **Calculation of Average PSNR Differences between RD curves (ITU-T SG16/Q6)**. VCEG. Marrakesh. 2001.

CHEN, C.-Y. et al. Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 16, n. 4, p. 553-558, Abril 2006.

CHEN, T.-C. et al. Single Reference Frame Multiple Current Macroblocks Scheme for Multi-Frame Motion Estimation in H.264/AVC. IEEE INT. SYMP. CIRCUITS AND SYSTEMS, Kobe, JAP, 2005. **Proceedings…** IEEE: New York, USA. p. 1790-1793.

CHEN, Y. et al. The Emerging MVC Standard for 3D Video Services. **EURASIP Journal on Advances in Signal Processing**, p. 1-13, 2009.

DODGSON, N. A. Autostereoscopic 3D Displays. **Computer**, v. 38, n. 8, p. 31-36, Agosto 2005.

FUJIFILM. FinePix REAL 3D W3 | Fuji Fulm Global, 2010. Disponivel em: <http://www.fujifilm.com/products/3d/camera/finepix_real3dw3/>. Acesso em: December 2012.

GONZALEZ, R.; WOODS, R. **Processamento de Imagens Digitais**. São Paulo: Edgard Blücher, 2003.

GRELLERT, M. et al. A Multilevel Data Reuse Scheme for Motion Estimation and Its VLSI Design. In: IEEE SYMPOSIUM ON CIRCUITS AND SYSTEM, Rio de Janeiro, BRA, 2011. **Proceedings…** IEEE: New York, USA, p. 583-586.

GUPTE, A. D. et al. Memory Bandwidth and Power Reduction Using Lossy Reference Frame Compression in Video Encoding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 21, n. 2, p. 225-230, February 2011.

HUFFMAN, D. A. A method for the Construction of Minimum-redundancy Codes. **IRE**, v. 40, n. 9, p. 1098-1101, 1952.

ISO. ISO - International Organization for Standardization. **ISO - International Organization for Standardization**, 2012. Disponivel em: <http://www.iso.org/iso/home.html>. Acesso em: 11 December 2012.

ITU-T. **ITU-T Rec. P.910 (09/99) Subjective video quality assesment methods for multimedia applications**. ITU-T. [S.l.], p. 37. 2007.

JCT-VC. **High efficiency video coding (HEVC) text specification draft 9 (JCTVC-K1003)**. JCT-VC. Shanghai. 2012.

JEDEC. DDR3 SDRAM STANDARD | JEDEC. **JEDEC**, 2010. Disponivel em: <http://www.jedec.org/standards-documents/docs/jesd-79-3d>. Acesso em: 20 Dezembro 2010.

JVT TEAM. **Draft ITU-T Rec. and final draft international standard of joint video specification**. [S.l.]: [s.n.]. 2003.

JVT TEAM. **Common Test Conditions for Multiview Video Coding**. Doc. JVT-T207. [S.l.]: [s.n.]. 2006.

JVT TEAM. **Editors' draft revision to ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding – in preparation for ITU-T SG 16 AAP Consent (in integrated form)**. Doc. JVT-AA07. [S.l.]: [s.n.]. 2009.

JVT TEAM. **Joint Multiview Vido Coding**. [S.l.]. 2010.

KONRAD, J.; HALLE, M. 3-D Disyplays and Signal Processing - An answer to 3-D ills? **IEEE Signal Processing Magazine**, v. 24, n. 6, p. 11-97, November 2007.

KUHN, P. **Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation**. Boston: Kluwer Academic Publishers, 1999.

MA, Z.; SEGALL, A. Frame Buffer Compression for Low-Power Video Coding. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, Brussels, BEL, 2011. **Proceedings…** IEEE: New York, USA, p. 757-760.

MAX, J. Quantizing for minimum distortion. **IRE Transactions on Information Theory**, v. 6, n. 1, p. 7-12, March 1960.

MERKLE, P. et al. Efficient Prediction Structures for Multiview Video Coding. **IEEEE Transactions on Circuits and Systems for Video Technology**, Piscataway, v. 17, n. 10, p. 14611473, November 2007.

MICRON. **TN-46-12: Mobile DRAM Power-Saving Features/Calculations**. [S.l.], p. 10. 2002.

MICRON. **256Mb DDR SDRAM (x4, x8, x16) Component Data Sheet**. [S.l.], p. 91. 2003.

MICRON. **512Mb: x4, x8, x16 DDR2 SDRAM**. [S.l.], p. 133. 2004.

MICRON. **1Gb: x4, x8, x16 DDR3 SDRAM**. [S.l.], p. 91. 2006.

MICRON. **1Gb: x16, x32 Mobile LPDDR SDRAM**. [S.l.], p. 95. 2007.

RICHARDSON, I. **H.264 and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia**. [S.l.]: John Wliey and Sons, 2003.

RODRIGUEZ, S.; JABOC, B. Energy/Power breakdown of pipelined nanometer caches (90nm/65nm/45nm/32). In: INTERNATIONAL SYMPOSIUM ON LOW POWER ELECTRONICS AND DESIGN, Tegernsee, GER, 2006. **Proceedings…** ACM: New York, USA, p. 25-30.

SCHWARZ, H.; MARPE, D.; WIEGAND , T. Analysis of hierarchical B pictures and MCTF. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA & EXPO, Toronto, CAN, 2006. **Proceedings…** IEEE: New York, USA.

SESHADRINATHAN, K. et al. Study of Subjective and Objective Quality Assessment of Video. **IEEE Transactions on Image Processing**, v. 19, n. 6, p. 1427-1441, June 2010.

SHAFIQUE, M. et al. Adaptive Power Management of On-Chip Video Memory for Multiview Video Coding. In: ACM/IEEE/EDA DESIGN AUTOMATION CONFERENCE, San Francisco, USA, 2012. **Proceedings…** IEEE: New York, USA. p. 866-875.

SHAFIQUE, M. et al. **Challenges and Joint Algorithm-Architecture Solutions for Emerging Multiview Video Coding**. KIT/UFRGS. Porto Alegre. 2012.

SHARP. Lynx 3D SH-03C, 2011. Disponivel em: <http://www.sharp.co.jp/products/sh03c/index.html>. Acesso em: December 2012.

SILVEIRA, D. et al. Memory bandwidth reduction in video coding systems through context adaptive lossless reference frame compression. In: SOUTHERN PROGRAMMABLE LOGIC CONFERENCE, Bento Gonçalves, BRA, 2012. **Proceedings…** IEEE: New York, p. 1-6.

SINGH, H. et al. Enhanced leakage reduction techniques using intermediate strengs power gating. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 15, n. 11, p. 1215-1224, November 2007.

SMOLIC, A. et al. Coding Algorithms for 3DTV - A Survey. **IEEE Transactions on Circuits and Systems for Video Technology**, Piscataway, v. 17, n. 11, p. 1606-1621, November 2007.

SMOLIC, A.; KAUFF, P. Interactive 3-D video representation and coding technologies. **Proceedings of the IEEE**, v. 93, n. 1, p. 98-110, January 2005.

SONG, L. et al. An adaptive bandwidth reduction scheme for video coding. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS. Paris, FRA, 2010. **Proceedings…** IEEE: New York, USA, p. 401-404.

SULLIVAN, G.; WIEGAND, T. Rate-Distortion Optimizatoin for Video Compression. **IEEE Signal Processing Magazine**, v. 15, p. 74-90, 1998.

TSUNG, P.-K. et al. System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, New Orleans, USA, 2007. **Proceedings…** IEEE: New York, USA, p. 1001-1004.

TUAN, J.-C.; CHANG, T.-S.; JEN, C.-W. On the Data Reuse and Memory Bandwidth Analysis for Full-Search Block-Matching VLSI Architecture. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 12, n. 1, p. 61-72, Janeiro 2002.

VETRO, A.; WIEGAND, T.; SULLIVAN, G. Overview of the Stereo and Multiview Video Coding Extensions of the H.264/AVC MPEG-4 AVC Standard. **Proceedings of the IEEE**, v. 99, n. 4, p. 626-642, April 2011.

WANG, Z. et al. Memory efficient lossless compression of image sequences with JPEG-LS and temporal prediction. In: PICTURE CODING SYMPOSIUM, Krakov, POL, 2012. **Proceedings…** IEEE: New York, USA, p. 305-3008.

WIEGAND, T. et al. Overview of the H.264/AVC Video Coding Standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 13, n. 7, p. 560-576, Julho 2003.

WIEGAND, T.; SULLIVAN, G. J. Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard. **Proceedings of the IEEE**, v. 99, n. 4, p. 626-642, April 2011.

XIU-LI, T.; SHENG-KUI, D.; CAN-HUI, C. An analysis of TZSearch algorithm in JMVC. In: INTERNATIONAL CONFERENCE ON GREEN CIRCUITS AND SYSTEMS, Shangai, CHI, 2010. **Proceedings…** IEEE: New York, USA, p. 516-520.

YAN, X. et al. Depth map generation for 2D-to-3D conversion by limited user inputs and depth propagation. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Beijing, CHI, 2011. **Proceedings…** IEEE: New York, USA, p. 1-4.

YANG, S.; WOLF, W.; VIJAYKRISHNAN, N. Power and performance analysis of motion estimation based on hardware and software realizations. **IEEE Transactions on Computers**, v. 54, n. 6, p. 714-726, 2005.

ZATT, B. et al. A low-power memory architecture with application-aware power management for motion & disparity estimation in Multiview Video Coding. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, San Jose, USA, 2011. **Proceedings…** IEEE: New York, USA, p. 40-47.

ZATT, B. et al. Run-Time Adaptive Energy-Aware Motion and Disparity Estimation in Multiview Video Coding. In: DESIGN AND AUTOMATION CONFERENCE, San Diego, USA, 2011. **Proceedings…** IEEE: New York, USA, p. 1026-1031.

ZHEN, L. et al. View-Adaptive Motion Estimation and Disparity Estimation for Low Complexity Multiview Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 20, n. 6, p. 925-930, June 2010.

APPENDIX A – MULTIVIEW VIDEO TEST SEQUENCES

This appendix presents the description of the multiview test sequences used in the experiments of this Master Thesis. Three index were calculated to measure characteristics of each individual video sequence: Spatial Index (SI), Temporal Index (TI) and Disparity Index (DI) (ITU-T, 2007). Tables A.1, A.2 and A.3 present the SI/TI/DI values for all considered multiview videos.

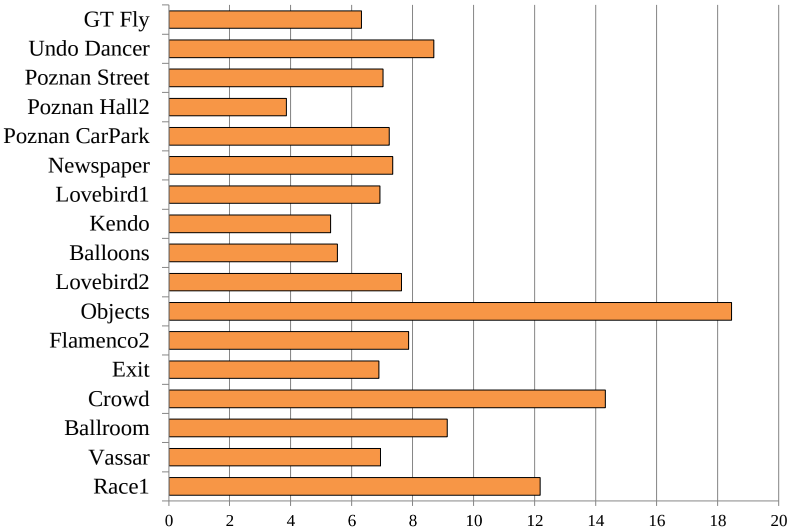


Figure A.1: Spatial Index (SI) comparison.

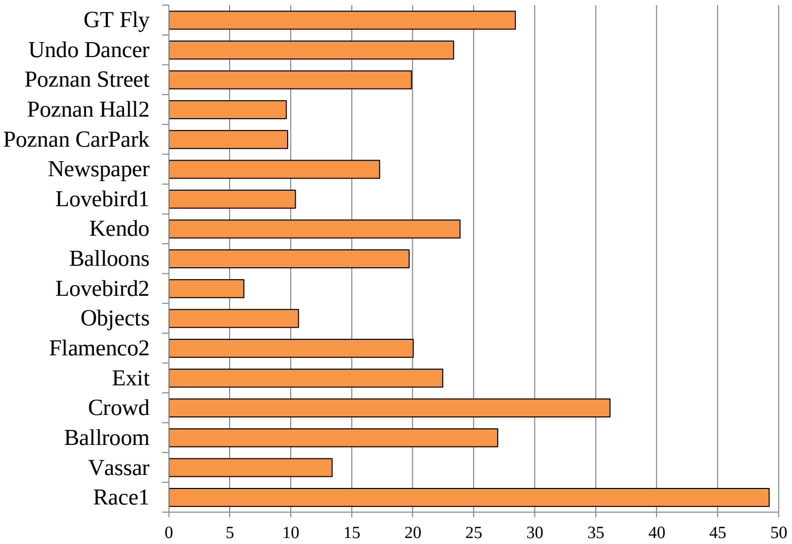


Figure A.2: Temporal Index (TI) comparison.

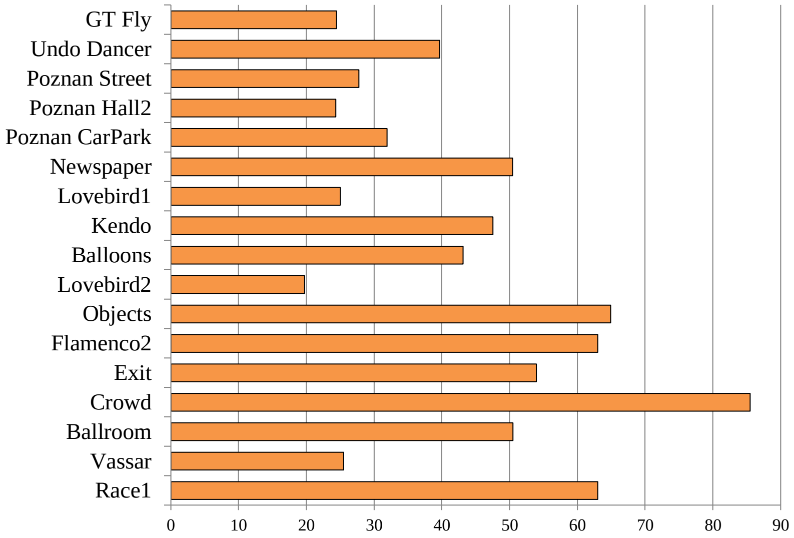


Figure A.3: Disparity Index (DI) Comparison.

APPENDIX B – STATISTIC HUFFMAN TABLES, NON-UNIFORM QUANTIZATION LEVELS AND THRESHOLDS DEFINITION

B.1 Off-line Static Search Map for ME/DE

Table B.1: DE off-line *StatMap* for the proposed power-gating control (x1000).

SW_V (#MB)	***SW_H (#MB)***												
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	34.7	39.4	40.4	41.6	42.6	43.7	43.7	43.7	42.6	41.5	40.4	39.2	30.7
1	36.8	41.6	42.8	43.9	45.1	46.2	46.2	46.2	45.0	43.9	42.7	41.5	32.5
2	39.1	44.4	45.6	46.8	48.0	49.4	50.5	49.2	48.0	46.7	45.4	44.2	34.5
3	42.2	47.5	48.8	50.1	51.4	54.2	53.1	54.2	51.3	49.9	48.6	47.1	36.1
4	46.2	51.8	53.2	54.6	57.6	57.8	59.1	57.6	57.6	54.5	53.0	51.4	37.7
5	47.9	53.5	55.1	58.1	58.3	61.2	61.2	61.2	58.2	58.1	54.9	53.1	39.1
6	47.9	53.5	56.6	56.9	59.7	61.2	61.2	61.2	59.6	56.7	56.6	53.1	39.1
7	47.9	53.5	55.0	58.1	58.1	61.2	61.2	61.2	58.0	58.1	54.8	53.1	39.1
8	46.1	51.7	53.2	54.6	57.7	57.7	59.2	57.5	57.7	54.4	52.9	51.3	37.9
9	44.5	49.9	51.3	52.7	54.0	57.1	55.7	57.1	53.8	52.5	51.1	49.5	36.8
10	42.8	48.1	49.5	50.7	52.0	53.4	55.1	53.2	51.9	50.6	49.2	47.7	35.7
11	39.0	41.1	42.2	43.3	44.4	45.5	45.5	45.5	44.3	43.1	41.9	40.6	31.0
12	32.2	33.8	34.7	35.6	36.5	37.4	37.5	37.4	36.4	35.3	34.3	33.3	25.3
μ=48.2 / σ=4.1													

Table B.2: ME off-line *StatMap* for the proposed power-gating control. (x1000)

SW_V (#MB)	***SW_H (#MB)***												
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	22.9	28.2	28.9	29.4	30.0	30.4	30.4	30.4	30.4	30.3	30.1	29.6	22.2
1	24.4	30.4	31.1	31.7	32.3	32.8	32.8	32.8	32.7	32.6	32.4	31.9	23.5
2	24.5	30.5	31.3	31.9	32.5	37.4	141.6	33.0	32.9	32.8	32.6	32.1	23.7
3	24.7	30.7	31.5	32.1	32.7	147.1	46.3	147.1	33.1	33.1	32.9	32.3	23.7
4	25.2	31.2	32.0	32.7	149.3	46.9	153.1	40.8	149.3	33.6	33.4	32.8	23.8
5	25.3	31.4	34.8	150.4	44.9	158.4	158.4	158.4	45.2	150.5	36.7	33.0	23.9
6	25.3	31.4	146.5	44.0	154.4	158.4	158.4	158.4	154.4	45.1	146.5	33.0	23.9
7	25.3	31.4	32.1	150.4	40.6	158.4	158.4	158.4	41.3	150.5	33.6	33.0	23.9
8	24.2	30.0	30.8	31.5	149.3	45.2	153.1	39.7	149.3	32.4	32.2	31.7	22.9
9	22.8	28.4	29.1	29.7	30.3	147.8	42.9	147.8	30.7	30.7	30.5	30.0	21.7
10	21.1	26.3	27.0	27.6	28.2	32.7	142.5	28.6	28.6	28.5	28.4	27.9	20.2
11	18.5	22.5	23.1	23.7	24.2	24.6	24.6	24.6	24.5	24.5	24.4	24.0	17.5
12	15.1	17.5	18.0	18.5	18.9	19.2	19.2	19.2	19.2	19.2	19.1	18.7	14.0
μ=50.2 / σ=4.7													

B.2 Static Quantization Levels and Huffman-Tables

Table B.3: Quantization level values: nLev=32.

#level	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
value	-23,86	-20,08	-17,42	-15,30	-13,49	-11,89	-10,44	-9,10
#level	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Value	-7,83	-6,64	-5,49	-4,38	-3,30	-2,24	-1,19	-0,16
#level	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
value	0,88	1,91	2,96	4,02	5,10	6,21	7,36	8,55
#level	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**21**
Value	9,82	11,16	12,61	14,21	16,02	18,14	20,80	24,58

Table B.4: Quantization level values: nLev=16.

#level	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
value	-19.449	-15.024	-11.811	-9.152	-6.804	-4.646	-2.602	-0.621
#level	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Value	1.340	3.322	5.365	7.523	9.871	12.530	15.743	20.169

Table B.5: Quantization level values: nLev=8.

#level	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
value	-14,70	-9,31	-5,14	-1,43	2,15	5,86	10,03	15,42

Table B.6: Huffman-table probabilities for lossless compression.

Residue Value	**Probability**	**Residue Value**	**Probability**	**Residue Value**	**Probability**	**Residue Value**	**Probability**
-128	3,91E-06	**-64**	3,10E-05	**0**	2,86E-01	**64**	4,02E-05
-127	3,58E-06	**-63**	3,21E-05	**1**	1,41E-01	**65**	3,83E-05
-126	3,58E-06	**-62**	3,17E-05	**2**	7,65E-02	**66**	3,77E-05
-125	3,58E-06	**-61**	3,18E-05	**3**	4,65E-02	**67**	3,35E-05
-124	4,80E-06	**-60**	3,40E-05	**4**	3,10E-02	**68**	3,10E-05
-123	4,38E-06	**-59**	4,03E-05	**5**	2,18E-02	**69**	2,82E-05
-122	3,53E-06	**-58**	4,19E-05	**6**	1,60E-02	**70**	3,19E-05
-121	4,90E-06	**-57**	4,77E-05	**7**	1,20E-02	**71**	2,96E-05
-120	4,14E-06	**-56**	4,95E-05	**8**	9,15E-03	**72**	2,84E-05
-119	3,91E-06	**-55**	5,05E-05	**9**	7,17E-03	**73**	2,59E-05
-118	4,14E-06	**-54**	5,54E-05	**10**	5,76E-03	**74**	2,37E-05
-117	3,16E-06	**-53**	5,84E-05	**11**	4,72E-03	**75**	2,32E-05
-116	3,91E-06	**-52**	6,54E-05	**12**	3,91E-03	**76**	2,32E-05
-115	4,38E-06	**-51**	6,66E-05	**13**	3,30E-03	**77**	2,24E-05
-114	3,20E-06	**-50**	6,84E-05	**14**	2,70E-03	**78**	1,91E-05
-113	5,37E-06	**-49**	7,61E-05	**15**	2,33E-03	**79**	1,83E-05
-112	3,58E-06	**-48**	7,70E-05	**16**	1,99E-03	**80**	1,82E-05
-111	3,39E-06	**-47**	8,69E-05	**17**	1,68E-03	**81**	1,72E-05
-110	3,91E-06	**-46**	8,90E-05	**18**	1,46E-03	**82**	1,74E-05
-109	3,67E-06	**-45**	9,65E-05	**19**	1,28E-03	**83**	1,54E-05
-108	3,72E-06	**-44**	1,02E-04	**20**	1,14E-03	**84**	1,61E-05
-107	4,85E-06	**-43**	1,10E-04	**21**	1,00E-03	**85**	1,43E-05
-106	3,39E-06	**-42**	1,23E-04	**22**	8,69E-04	**86**	1,40E-05
-105	4,29E-06	**-41**	1,26E-04	**23**	7,77E-04	**87**	1,37E-05

-104	5,04E-06	**-40**	1,41E-04	**24**	6,96E-04	**88**	1,15E-05
-103	4,19E-06	**-39**	1,52E-04	**25**	6,32E-04	**89**	1,10E-05
-102	6,40E-06	**-38**	1,57E-04	**26**	5,58E-04	**90**	1,05E-05
-101	6,40E-06	**-37**	1,67E-04	**27**	4,81E-04	**91**	1,02E-05
-100	6,03E-06	**-36**	1,93E-04	**28**	4,71E-04	**92**	9,47E-06
-99	6,55E-06	**-35**	2,00E-04	**29**	4,30E-04	**93**	8,85E-06
-98	6,03E-06	**-34**	2,25E-04	**30**	3,83E-04	**94**	9,09E-06
-97	5,51E-06	**-33**	2,39E-04	**31**	3,38E-04	**95**	1,00E-05
-96	5,09E-06	**-32**	2,58E-04	**32**	3,22E-04	**96**	8,15E-06
-95	6,97E-06	**-31**	2,88E-04	**33**	2,97E-04	**97**	9,51E-06
-94	6,12E-06	**-30**	3,07E-04	**34**	2,72E-04	**98**	7,11E-06
-93	6,97E-06	**-29**	3,38E-04	**35**	2,48E-04	**99**	6,59E-06
-92	7,86E-06	**-28**	3,69E-04	**36**	2,28E-04	**100**	6,73E-06
-91	6,03E-06	**-27**	4,11E-04	**37**	2,07E-04	**101**	7,82E-06
-90	7,35E-06	**-26**	4,47E-04	**38**	1,86E-04	**102**	8,24E-06
-89	8,29E-06	**-25**	5,02E-04	**39**	1,71E-04	**103**	6,12E-06
-88	9,00E-06	**-24**	5,71E-04	**40**	1,66E-04	**104**	6,64E-06
-87	8,34E-06	**-23**	6,34E-04	**41**	1,52E-04	**105**	6,59E-06
-86	8,81E-06	**-22**	7,08E-04	**42**	1,48E-04	**106**	5,42E-06
-85	1,05E-05	**-21**	8,04E-04	**43**	1,38E-04	**107**	5,70E-06
-84	1,00E-05	**-20**	9,18E-04	**44**	1,25E-04	**108**	5,37E-06
-83	1,17E-05	**-19**	1,05E-03	**45**	1,22E-04	**109**	6,26E-06
-82	1,21E-05	**-18**	1,21E-03	**46**	1,12E-04	**110**	4,95E-06
-81	1,18E-05	**-17**	1,37E-03	**47**	1,01E-04	**111**	4,52E-06
-80	1,19E-05	**-16**	1,64E-03	**48**	9,57E-05	**112**	3,25E-06
-79	1,36E-05	**-15**	1,90E-03	**49**	9,18E-05	**113**	3,96E-06
-78	1,36E-05	**-14**	2,29E-03	**50**	8,95E-05	**114**	4,00E-06
-77	1,30E-05	**-13**	2,71E-03	**51**	8,44E-05	**115**	4,43E-06
-76	1,56E-05	**-12**	3,23E-03	**52**	7,80E-05	**116**	4,38E-06
-75	1,57E-05	**-11**	4,01E-03	**53**	7,65E-05	**117**	4,05E-06
-74	1,73E-05	**-10**	4,91E-03	**54**	6,56E-05	**118**	4,76E-06
-73	1,79E-05	**-9**	6,14E-03	**55**	6,81E-05	**119**	3,39E-06
-72	1,87E-05	**-8**	7,80E-03	**56**	6,15E-05	**120**	3,67E-06
-71	2,09E-05	**-7**	1,01E-02	**57**	5,57E-05	**121**	3,16E-06
-70	2,29E-05	**-6**	1,34E-02	**58**	5,35E-05	**122**	4,19E-06
-69	2,43E-05	**-5**	1,82E-02	**59**	5,27E-05	**123**	3,30E-06
-68	2,52E-05	**-4**	2,57E-02	**60**	4,92E-05	**124**	3,86E-06
-67	2,53E-05	**-3**	3,78E-02	**61**	4,54E-05	**125**	3,91E-06
-66	2,84E-05	**-2**	5,95E-02	**62**	4,14E-05	**126**	4,80E-06
-65	2,99E-05	**-1**	9,82E-02	**63**	4,13E-05	**127**	4,10E-06

Table B.7: Huffman-table probabilities for lossy compression (*nLev=32*).

Residue Value	**Probability**	**Residue Value**	**Probability**	**Residue Value**	**Probability**	**Residue Value**	**Probability**
-16	0,007	**-8**	0,014	**0**	0,286	**8**	0,007
-15	0,002	**-7**	0,010	**1**	0,141	**9**	0,010
-14	0,003	**-6**	0,013	**2**	0,077	**10**	0,004
-13	0,003	**-5**	0,018	**3**	0,078	**11**	0,006
-12	0,004	**-4**	0,026	**4**	0,022	**12**	0,004
-11	0,006	**-3**	0,038	**5**	0,016	**13**	0,003
-10	0,004	**-2**	0,060	**6**	0,012	**14**	0,002
-9	0,005	**-1**	0,098	**7**	0,009	**15**	0,003

Table B.8: Huffman-table probabilities for lossy compression (*nLev=16*).

Residue Value	**Probability**	**Residue Value**	**Probability**
-8	0.010	**0**	0.427
-7	0.005	**1**	0.123
-6	0.010	**2**	0.053
-5	0.009	**3**	0.028
-4	0.024	**4**	0.016
-3	0.032	**5**	0.014
-2	0.064	**6**	0.008
-1	0158	**7**	0.007

Table B.9: Huffman-table probabilities for lossy compression (*nLev=8*).

Residue Value	**Probability**
-4	0.017
-3	0.017
-2	0.038
-1	0.141
0	0.602
1	0.099
2	0.050
3	0.017
4	0.018

APPENDIX C – RESUMO EM PORTUGUÊS

Hierarquia de Memória Eficiente em Energia para a Estimação de Movimento e de Disparidade da Codificação de Vídeo Multivistas

C.1 Resumo

Esta dissertação de mestrado propõe uma hierarquia de memória para a Estimação de Movimento e de Disparidade (ME/DE) centrada nas referências da codificação, estratégia chamada de *Reference-Centered Data Reuse* (RCDR), com foco em redução de energia em codificadores de vídeo multivistas (MVC - *Multiview Video Coding*). Nos codificadores MVC, a ME/DE é responsável por praticamente 98% do consumo total de energia. Além disso, até 90% desta energia está relacionada com a memória do codificador: (a) acessos à memória externa para a busca das referências da ME/DE (45%) e (b) memória interna (*cache*) para manter armazenadas as amostras da área de busca e enviá-las para serem processadas pela ME/DE (45%). O principal objetivo deste trabalho é minimizar de maneira conjunta a energia consumida pelo módulo de ME/DE com relação às memórias externa e interna necessárias para a codificação MVC. A hierarquia de memória é composta por uma memória interna (a qual armazena a área de busca inteira), um controle dinâmico para a estratégia de *power-gating* da memória interna e um compressor de resultados parciais. Um controle de buscas foi proposto para explorar o comportamento da busca com o objetivo de atingir ainda mais reduções de energia. Além disso, este trabalho também agrega à hierarquia de memória um compressor de quadros de referência de baixa complexidade. A estratégia RCDR provê reduções de até 68% no consumo de energia quando comparada com estratégias estado-da-arte que são centradas no bloco atual da codificação. O compressor de resultados parciais é capaz de reduzir em 52% a comunicação com memória externa necessária para o armazenamento desses elementos. Quando comparada a técnicas de reuso de dados que não acessam toda área de busca, a estratégia RCDR também atinge os melhores resultados em consumo de energia, visto que acessos regulares a memórias externas DDR são energeticamente mais eficientes. O compressor de quadros de referência reduz ainda mais o número de acessos a memória externa (2,6 vezes menos acessos), aliando isso a perdas insignificantes na eficiência da codificação MVC. A memória interna requerida pela estratégia RCDR é até 74% menor do que estratégias centradas no bloco atual, como Level C. Além disso, o controle dinâmico para a técnica de *power-gating* provê reduções de até 82% na energia estática, o que é o melhor resultado entre os trabalho relacionados. A energia dinâmica é tratada pela técnica de união dos blocos candidatos, atingindo ganhos de mais de 65%. Considerando as reduções de consumo de energia atingidas pelas técnicas propostas neste trabalho, conclui-se que o sistema de hierarquia de memória proposto nesta dissertação atinge seu

objetivo de atender às restrições impostas pela codificação MVC, no que se refere ao processamento do módulo de ME/DE.

C.2 Introdução e Motivação

O padrão estado da arte em codificação de vídeos multivistas (MVC – *Multiview Video Coding*) (JVT TEAM, 2009) provê uma eficiência de codificação entre 20% e 50% maior quando comparado com o padrão H.264/AVC utilizando a estratégia *simulcast* (MERKLE, SMOLIC, *et al.*, 2007). Além de novos elementos sintáticos para suportar a representação de vídeos multivistas, a principal inovação dos codificadores MVC é a predição inter-vistas, a qual utiliza a Estimação de Disparidade (DE – *Disparity Estimation*) para capturar o deslocamento de posição dos objetos em razão das diferentes posições das câmeras. Juntamente com a Estimação de Movimento (ME – *Motion Estimation*), a DE consome a maior parte da energia gasta pelos codificadores MVC (mais de 92%) (ZATT, SHAFIQUE, *et al.*, 2011).

A ME/DE é usada para buscar uma região da imagem (bloco candidato) que apresenta o melhor casamento dentro de um ou mais quadros de referência (quadros já codificados). A busca é realizada dentro de uma área de busca utilizando um algoritmo de busca, como o TZ Search (XIU-LI, SHENG-KUI e CAN-HUI, 2010). Esta área de busca é tipicamente acessada pela memória externa e armazenada na memória interna. Mesmo quando algoritmos rápidos são utilizados, acessos à memória frequentes e grande memória interna são requeridos e, por conseguinte, um alto consumo de energia é verificado. Além disso, uma vez que 90% da energia consumida pela ME/DE são gastas com a memória do codificador, a redução da energia das memórias interna e externa é mandatória para atender as restrições e requerimentos de projeto impostos pelos dispositivos portáteis da atualidade (ZATT, SHAFIQUE, *et al.*, 2011).

Inúmeros trabalhos já publicados na literatura têm como objetivo lidar com as penalidades relacionadas à memória no processamento do módulo de ME/DE na codificação MVC. Alguns destes trabalhos propõem estratégias de armazenar localmente as amostras de referência da ME/DE. Estes trabalhos podem ser classificados de dois tipos: (1) estratégias de reuso de dados centradas no macrobloco (MBDR – *MB-Centered Data Reuse*) (CHEN, HUANG, *et al.*, 2006) e (2) estratégias de reuso de dados centradas na referência (RCDR – *Reference-Centered Data Reuse*) (TSUNG, DING, *et al.*, 2007). Entretanto, as estratégias MBDR sofrem com o grande número de quadros de referência e grandes áreas de buscas requeridas pela codificação MVC. Enquanto isso, os trabalhos já publicados baseados na estratégia RCDR não consideram o impacto das penalidades geradas (resultados parciais de codificação precisam ser salvos) no tráfego com a memória e no tamanho da memória interna. Desta maneira, um dos objetivos do nosso trabalho é desenvolver uma técnica de reuso de dados, baseada na estratégia RCDR, a qual é mais promissora em termos de redução de energia, o qual se preocupa com as penalidades de armazenamento e transmissão dos resultados parciais da codificação.

Outro conjunto de trabalhos visa a reduzir os acessos à memória externa aplicando técnicas de compressão sobre as amostras de referências antes que elas sejam armazenadas na memória externa. Neste caso, a ME/DE necessita primeiro recuperar as amostras originais (utilizando o caminho de descompressão) para que possa ter as referências para o processo de busca. Técnicas de compressão sem perdas foram propostas em (SILVEIRA, GRELLERT, *et al.*, 2012) e (WANG, CHANDA, *et al.*, 2012). Compressões com perdas baseadas no uso de quantizadores foram propostas em

(MA e SEGALL, 2011) e (GUPTE, AMRUTUR, *et al.*, 2011). Uma adaptação ao conteúdo do vídeo foi primeiramente explorada em (SONG, ZHOU, *et al.*, 2010). Entretanto, tais estratégias não foram desenvolvidas para as restrições impostas pela codificação MVC, as quais requerem muito mais acessos à memória para o módulo de ME/DE. Além disso, as estruturas de predição MVC apresentam muitas dependências entre quadros vizinhos (temporais e de disparidade), fazendo com que o erro seja propagado ao longo do GOP (*Group of Pictures*) em todas as direções de busca. Assim, outro objetivo do nosso trabalho é comprimir os dados de referência (diminuindo o número de acessos à memória externa) minimizando a propagação de erro ao longo da estrutura de predição MVC.

Trabalhos anteriores do grupo também já se preocuparam em reduzir a energia consumida pelas memórias interna/externa durante o processamento dos módulos de ME/DE da codificação MVC. Em (ZATT, SHAFIQUE, *et al.*, 2011), é apresentada uma arquitetura para a ME/DE de baixo consumo de energia que utiliza o conceito de mapa de busca e área de busca dinâmica. O trabalho (ZATT, SHAFIQUE, *et al.*, 2011) estendeu o trabalho anterior empregando técnicas de múltiplos estados de potência no controle de power-gating dinâmico para reduzir o consumo de energia estático. Mais recentemente, em (SHAFIQUE, ZATT, *et al.*, 2012) é proposto um gerenciamento de energia da memória interna baseada em técnicas como eliminação da direção de busca, visando reduzir ainda mais o consumo de energia. Estas técnicas são todas baseadas em algoritmos dinâmicos que acessam a memória externa de maneira irregular e ocasionam *misses* na memória interna. Com isso em mente, este trabalho tem como objetivo proporcionar um padrão de acessos regulares à memória externa, o que é energeticamente mais eficiente. Para isso, este trabalho acessa e armazena uma área de busca inteira. Além disso, são propostas técnicas para reduzir o consumo de energia da memória interna.

Este trabalho propõe uma hierarquia de memória para o módulo de ME/DE focando na codificação MVC com o objetivo de reduzir o consumo de energia ocasionado pelo armazenamento das amostras de referência na memória interna e pelo acesso dessas amostras na memória externa do codificador. A hierarquia de memória é composta por uma memória interna, um controla dinâmico para o power-gating, um compressor de resultados parciais e um compressor de quadros de referência adaptativo ao conteúdo do vídeo. Adicionalmente, um controle de busca customizado é proposto para explorar o comportamento da busca com o objetivo de reduzir ainda mais o consumo de energia.

Este texto está organizado com ao seguir: Seção C.2 apresenta a hierarquia de memória baseada nas referências da codificação e todas as técnicas propostas em detalhes; Seção C.3 mostra e discute os resultados e comparações com trabalhos relacionados; finalmente, Seção C.4 conclui o texto.

C.3 Hierarquia de Memória Proposta

Figura C.1 apresenta os módulos que compõem a arquitetura da nossa hierarquia de memória proposta neste trabalho.

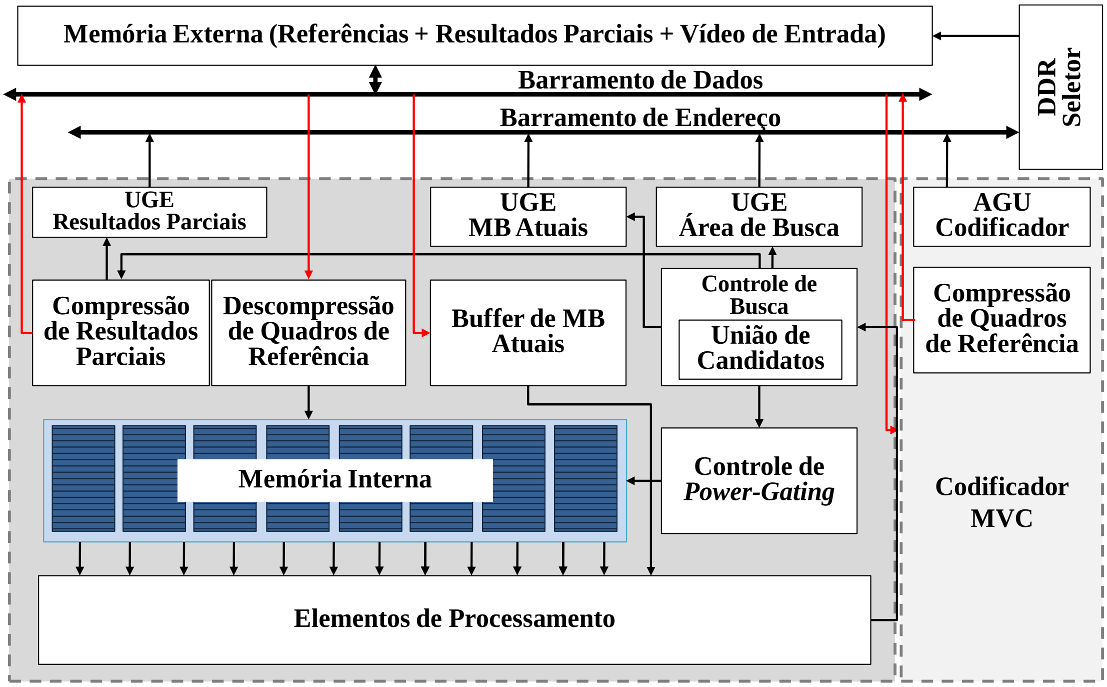


Figura C.1: Diagrama em blocos da hierarquia memória.

Para controlar a busca da ME/DE e, consequentemente, o padrão de acessos, um Controle de Busca é definido. Inicialmente, o Controle de Busca envia requerimentos para o acesso da área de busca na memória externa. Estes acessos são representados no formato de posições nativas do vídeo, coordenadas *x* e *y*. Então, as unidades de gerenciamento de endereços (UGEs) são utilizadas para traduzir os acessos em múltiplas posições físicas de memória. Antes que sejam armazenados na memória interna, as amostras da área de busca devem ser descomprimidas de modo a serem recuperadas como as originais (com ou sem perdas). Uma rajada de posições de blocos candidatos é gerada pelo Controle de Busca de acordo com o algoritmo TZ Search. Estas posições são rearranjadas pela unidade de união dos blocos candidatos com o objetivo de reduzir o número de acessos à memória interna. O controle de power-gating monitora as estatísticas da busca e defini o estado de energia das linhas de memória de maneira apropriada. Os candidatos são processador por um conjunto de elementos de processamento (não descritos neste trabalho). O melhor casamento e o seu custo de SAD são passados para o Controle de Busca. Em razão do processamento fora de ordem da estratégia RCDR, os vetores de movimento/disparidade e os custos de SAD temporários devem ser salvos para serem posteriormente utilizados pelo modo de decisão do codificador MVC. Estes resultados parciais são comprimidos usando quantizadores não-lineares e codificadores de Huffman. Uma vez que o resultado da compressão possui comprimento variável, esses dados são enviados para a memória externa uma vez que o buffer local está completo. Uma UGE específica foi desenvolvida para esta transmissão.

C.3.1 Reuso de Dados Centrado em Referências

A estratégia RCDR utilizada uma lógica de dependência inversa entre quadros de referência e MB atuais. Neste esquema, a área de busca é acessada da memória externa e todos aqueles MBs que necessitam dessas informações de referência são processados no mesmo instante. Em outras palavras, as informações de referência ‘chama’ os MBs para serem processados. Por esta razão, nós definimos o termo *quadros dependentes* para aqueles quadros que são ‘chamados’ por um dado quadro de referência.

A Figura C.2 apresenta as diferenças entre as estratégias RCDR e MBDR. É possível observar que na estratégia MBDR todos os MB atuais necessitam de até quatro áreas de busca requerendo 4x mais memória interna para armazenamento. Adicionalmente, como aquelas áreas de busca são acessadas mais vezes no futuro (para a codificação de outros quadros), é necessária a retransmissão dessas amostras de referência. Em contrapartida, o armazenamento interno de apenas uma área de busca é necessário para o RCDR, resultando numa memória interna mais reduzida. Além disso, no RCDR a área de busca é acessada e lida da memória externa apenas uma vez. Na verdade, MBs atuais que pertencem a quadros dependentes são acessados múltiplas vezes. Entretanto, isso representa um pequeno impacto para ambas memórias interna e externa, como demonstrado na seção de resultados. O GDV (*Global Disparity Vector*) é levado em conta para localizar as posições dos MBs atuais para a DE, como mostrado na Figura C.2 .

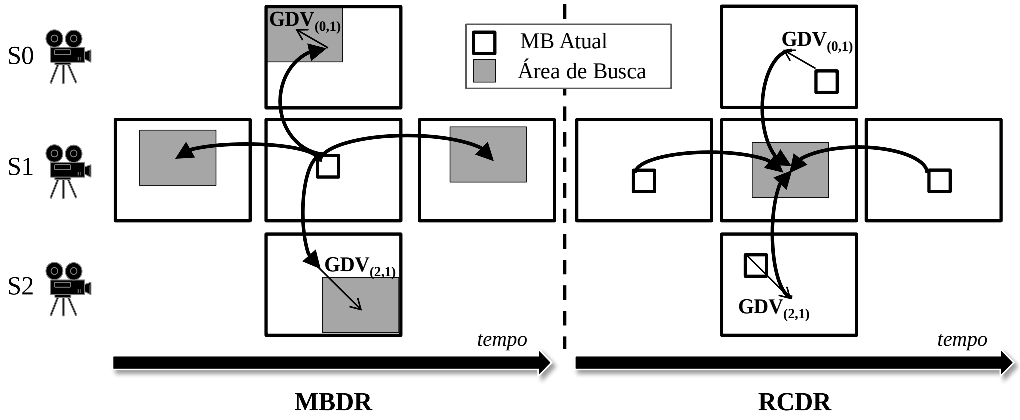


Figura C.2: (a) MBDR versus (b) RCDR.

C.3.2 Compressor de Resultados Parciais

Os resultados parciais são compostos por dois diferentes tipos de dados: (i) vetores de movimento/disparidade e (ii) custos de SAD. Estes dois tipos apresentam intervalos numéricos e comportamentos estatísticos diferentes. Por esta razão, iremos discuti-los de maneira separada.

De modo a concentrar estes vetores em um intervalo reduzido, um preditor espacial, (utilizando os vetores dos MBs acima e à esquerda) foi proposto. A distribuição dos vetores diferenciais é concentrada em um intervalo pequeno centrado no valor DVx=0. Para 54 valores que representam $\mu\pm2\sigma=95.8\%$ das ocorrências, uma tabela Huffman foi gerada de acordo com as técnicas apresentadas em (HUFFMAN, 1952). As coordenadas dos vetores diferencias fora deste intervalo são representadas por um código Huffman especial seguido pelo valor do vetor em uma representação binária de 8 bits, sem o uso de compressão.

Os valores de SAD são espalhados ao longo do grande intervalo numérico de representação. Para tal distribuição, uma etapa de quantização é necessária. Para reduzir o impacto dos erros de quantização, nós empregamos uma quantização não-linear desenvolvida para uma distribuição Gaussiana de acordo com o algoritmo Lloyd e o refinamento Lloyd-Max (MAX, 1960). O quantizador emprega 512 níveis otimizadas para o mínimo erro quadrático (MMSE – minimum mean squared error). Depois da quantização, os SADs quantizados são codificados usando uma tabela Huffman de 189 entradas.

C.3.3 Compressor de Quadros de Referência

A estratégia proposta comprime as amostras depois de elas serem completamente codificadas e reconstruídas (i.e., após o Filtro de Deblocagem). Neste ponto, as amostras reconstruídas são armazenadas na memória externa para serem futuramente utilizadas como referência para a ME/DE dos quadros subsequentes.

O nosso esquema é aplicado para cada bloco 4x4 do MB reconstruído. Inicialmente, uma predição intra-quadro simplificada utilizando apenas blocos de tamanho 4x4 é realizada para eliminar as redundâncias espaciais presentes nos quadros reconstruídos. De modo a evitar computação extra, a estratégia proposta aproveita o melhor modo 4x4 escolhido pelo modo de decisão do codificador MVC. É possível notar que nosso compressor usar o melhor modo da predição intra-quadro independentemente do modo escolhido para codificar o MB (que pode ser intra 16x16 ou inter-quadros/vistas). A predição intra-quadro simplificada em nosso esquema é compatível com a definição do padrão H.264/AVC para os blocos 4x4: 9 modos possíveis utilizando treze amostras vizinha, quando disponíveis. Então, o resíduo (diferença entre as amostras reconstruídas e preditas) é calculado. A distribuição dos valores de resíduo é muito mais concentrada quando comparada com a dos dados reconstruídos. Explorando esta distribuição mais concentrada, um codificador de entropia baseado em tabelas Huffman é aplicado. Uma vez que a predição intra-quadro explora as correlações espaciais da imagem, os blocos heterogêneos (texturizados) tendem a gerar uma distribuição mais espalhada dos valores do resíduo, o que não é desejável para o codificador Huffman. Para melhor lidar com tais blocos, o esquema proposto implementa uma quantização não linear para minimizar ainda mais o intervalo de representação dos dados.

Após a quantização, as amostras iniciais não podem ser identicamente recuperadas em razão da discretização do intervalo, causando perdas nos resultados de eficiência (relação taxa-distorção) do codificador MVC. Considerando o alto número de dependências nas estruturas de predição (domínios temporais e de disparidade), esses erros podem ser propagados durante todas as operações de ME/DE dentro do GOP. Para lidar com este problema, nós propomos uma estratégia de adaptação ao conteúdo para modificar dinamicamente a tabela Huffman e o passo de quantização de acordo com as características da imagem.

A nossa estratégia de adaptação ao conteúdo classifica os blocos 4x4 em quatros grupos de homogeneidade *HG={G0,G1,G2,G3}*. A classificação é feita utilizando a variância estatística das amostras originais do bloco como métrica de homogeneidade. Como é possível notar, o uso das amostras originais para a classificação dos blocos não insere nenhuma dependência de dados com etapas da codificação. Quatro diferente quantizadores não lineares foram definidos: *nLev(G0)=8, nLev(G1)=16, nLev(G2)=32* e *nLev(G3)=256*, onde nLev representa o número de intervalos de quantização (níveis). Eles foram adaptados para atingir a melhor eficiência possível (minimização conjunta do erro e do resíduo) para as propriedades específicas de cada grupo. O codificador Huffman é composto de quatro tabelas com: 8 entradas para o grupo G0, 16 entradas para o grupo G1, 32 entradas para o grupo G2 e 256 entradas para o grupo G3.

C.3.4 Organização da Memória Interna

A memória interna é logicamente definida como um buffer circular organizado como uma matriz de duas dimensões para prover mapeamento direto com os dados do vídeo. A organização é composta por B bancos de memória lógicos que rotacionam após cada passo de busca com o objetivo de evitar a retransmissão das amostras da área de busca já acessadas para os blocos anteriores (reuso de dados). Esta organização, entretanto, não é eficiente para a implementação física uma vez que a ME/DE necessita de leituras altamente paralelas dos MBs. A organização física é composta de 16 bancos de SRAM paralelos com palavra de 128 bits (16 amostras por linha). Cada linha memória armazena e alimenta um MB completo da área de busca em paralelo. Cada banco é adicionalmente dividido em setores de *n* linha representando uma coluna da área de busca. O número total de linhas é definido pelo número de MBs em uma área de busca. Nota-se que diferentemente da organização lógica, colunas de MBs não são deslocadas a cada passo de busca. Nesse evento, os setores da memória são renomeados de maneira apropriada.

Nós propomos uma estratégia de controle estatístico de power-gating que emprega múltiplos estados de potência com o objetivo de reduzir o consumo de energia estática. Este esquema não requer a extração de propriedades da imagem ou predição de acessos a memória em nível de MB para proporcionar uma solução simples e eficiente. Quatro estados são implementados (SINGH, SYLVESTER, *et al.*, 2007): *S0=OFF (Vdd=0), S1= Data Retentive (Vdd=Vdd*0.3), S2= Data Retentive (Vdd=Vdd*0.5)* e *S3= ON (Vdd= Vdd).* Onde cada estado tem uma energia de wakeup associada ($WE_{S0}> WE_{S1}> WE_{S2}>WE_{S3}=0$). Por esta razão, regiões que são frequentemente acessadas são mapeadas para o estado S3, regiões não utilizadas para o estado S0, e outras regiões são mapeadas para os estados S1-S2 de acordo com as estatísticas de uso extraídas em tempo de execução.

Embora a energia estática tenha se tornado dominante em memórias de altíssima densidade, a redução da energia dinâmica contribui significantemente com o consumo da energia total. Para evitar o frequente gasto de energia com as transições do sinal das linhas da memória interna, nós também definimos uma estratégia de união dos blocos candidatos da busca. Uma vez que múltiplos MBs dependentes são buscados simultaneamente na mesma área de busca, múltiplos pontos de acesso são requisitados mais de uma vez. Nossa união de blocos candidatos recebe todos os pontos de busca gerados pelo controle de busca e rearranja-os com o objetivo de processar ao mesmo tempo blocos candidatos repetidos. Deste modo, muitas transições das linhas da memória interna são evitadas. Para cada acesso salvo, temos uma redução no consumo de energia dinâmica.

C.4 Resultados e Discussões

Os resultados experimentais foram gerados utilizando sequências de vídeos reais e configurações do codificador de acordo com o JVT, órgão padronizador da codificação MVC (JVT TEAM, 2006). Um simulador de energia foi desenvolvido no nosso grupo de pesquisa foi utilizado para medir o consumo de energia da nossa hierarquia de memória e compará-lo com as soluções propostas em trabalhos relacionados. As energias estática e de wakeup das células de SRAM foram calculadas com base em (SINGH, SYLVESTER, *et al.*, 2007). Os ganhos de consumo de energia da memória externa foram avaliados utilizando a memória Low-Power DDR (LPDDR) MT46H64M16LF, com capacidade para 1 Gigabit de armazenamento (MICRON, 2007). Importante notar que os resultados experimentais já contêm as energias de wakeup das células de memória.

Figura C.3 apresenta os ganhos em consumo de energia com relação a memória externa considerando várias resoluções e diversos tamanhos de área de busca. A análise foi realizada para quatro diferentes cenários, em comparação com a estratégia tradicional MBDR Level-C (CHEN, HUANG, *et al.*, 2006). Observa-se que os ganhos de energia são cada vez maiores quanto for maior o número de vistas codificado. O mesmo comportamento foi verificado quanto maior for o tamanho da área de busca. Adicionalmente, a nossa estratégia não sofre perdas quando vídeos de maior resolução são codificados. Os maiores ganhos são verificados quando a estrutura de predição IBP é utilizada, visto que haverá um maior reuso da área de busca (mais MBs usarão a mesma área de busca). Estes resultados já incluem a redução no número de acessos à memória para o tratamento dos resultados parciais da codificação (tratados pelo caminho de compressão dos resultados parciais). O nosso compressor atinge, em média, uma redução de 53,2% na comunicação com a memória externa para o salvamento dos resultados parciais (comparado com o cenário onde não há este estágio de compressão).

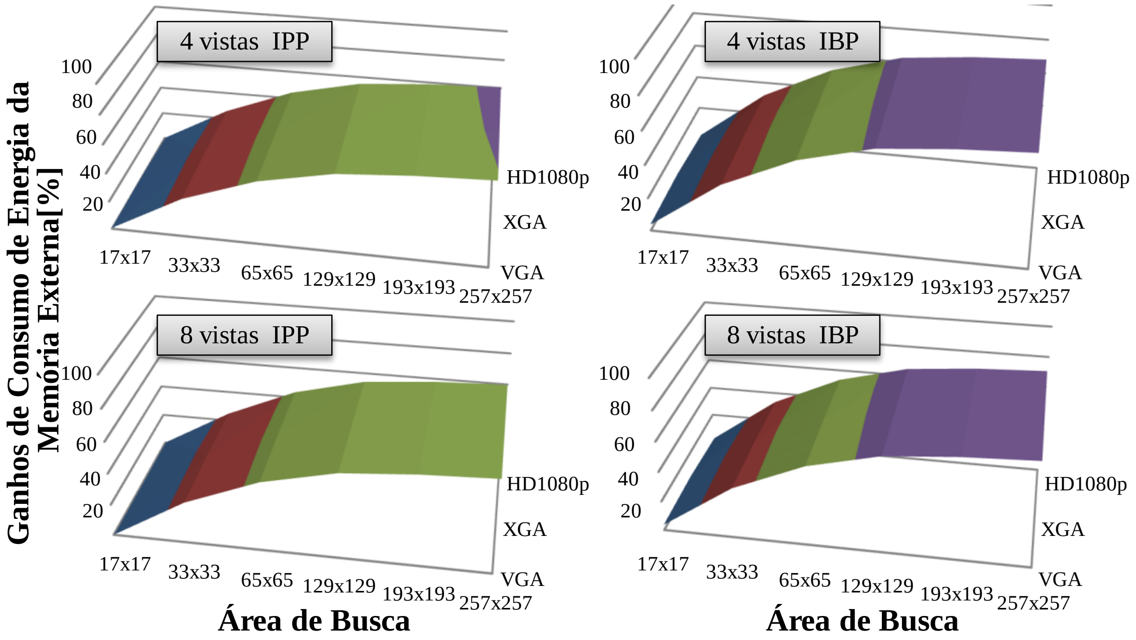


Figura C.3: Ganhos do consumo de energia da memória externa.

Quando a estratégia RCDR é comparada com soluções que não acessam a memória de maneira regular, como o trabalho (ZATT, SHAFIQUE, *et al.*, 2011), a nossa técnica é capaz ainda assim de reduzir o consumo de energia em cerca de 30%. A energia efetiva gasta com a leitura dos dados da memória LPDDR é 2,4x maior na estratégia RCDR, o que é esperado uma vez que toda a área de busca é acessada da memória externa. Entretanto, o padrão irregular de acessos da solução proposta em (ZATT, SHAFIQUE, *et al.*, 2011) lida com um alto consumo de energia em razão da constante ativação/desativação das páginas da memória LPDDR e, consequentemente, leituras em rajadas não são praticamente exploradas. Em termos de ganhos absolutos, a energia de ativação de páginas é reduzida em mais de 90% pela estratégia RCDR. A energia de ativação de páginas é dominante quando comparada à energia consumida pela solução de (ZATT, SHAFIQUE, *et al.*, 2011). Deste modo, mesmo que a estratégia RCDR necessite de um número maior de operações de leitura da memória LPDDR, o consumo de energia total relacionada à memória externa é reduzida em relação ao trabalho relacionado.

Comparado com o Level-C, o tamanho da memória interna necessário para a estratégia RCDR é altamente reduzido (veja na Figura C.4) uma vez que não há a necessidade de armazenar de maneira simultânea múltiplas áreas de busca. Nota-se que a nossa memória interna cresce de maneira bem mais suave com o aumento da área de busca. Além disso, comparado com o custo de armazenando das áreas de busca, o custo de armazenar os MBs atuais dependentes é praticamente insignificantes (este custo está contado na análise da Figura C.4). Este custo é mais amortizado quanto maior for o tamanho da área de busca. O tamanho reduzido da memória interna acarreta diretamente em ganhos de consumo de energia estática, como mostrado na Figura C.5. Comparado com o trabalho (CHEN, HUANG, *et al.*, 2006), 77% de redução de energia é alcançado sem o emprego da nossa técnica de power-gating. Quando o power-gating é acionado, a redução chega a ser de 88%, superando o trabalho (ZATT, SHAFIQUE, *et al.*, 2011) em 66%. A análise de energia considerou o cenário de 4 vistas e IBP.

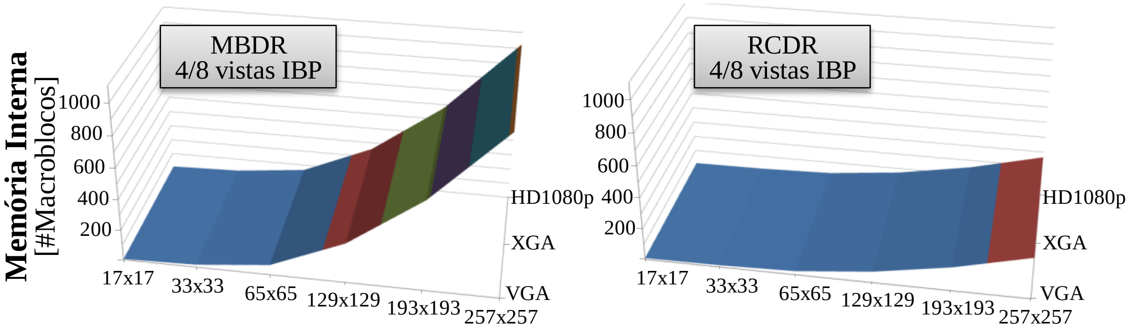


Figura C.4: Redução do tamanho da memória interna.

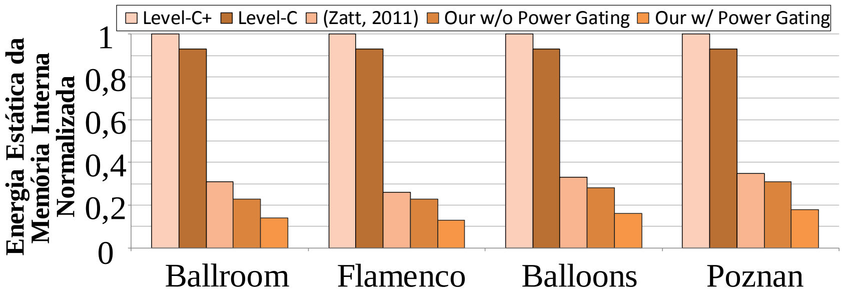


Figura C.5: Ganhos do consumo de energia estático da memória interna.

Em termos de redução da energia dinâmica da memória interna, a nossa estratégia de união dos blocos candidatos da ME/DE reduz o consumo de energia em 65%, em média. Até onde os autores deste trabalho sabem, este é o primeiro trabalho da literatura que se preocupa também em reduzir esta parcela da energia interna.

Tabela C.1 apresenta uma comparação da nossa estratégia de compressão de quadros de referência adaptativa ao conteúdo com o estado da arte (MA e SEGALL, 2011) (GUPTE, AMRUTUR, *et al.*, 2011) (SONG, ZHOU, *et al.*, 2010) (SILVEIRA, GRELLERT, *et al.*, 2012). É possível notar que nenhum dos trabalhos avaliou as suas técnicas considerando a codificação MVC. Neste sentido, a propagação de erro ao longo das referências em cascata da ME/DE é significantemente menor nas avaliações dos trabalhos relacionados.

Tabela C.1: Resultados e Comparação do Compressor de Quadros de Referência

Parâmetro	**Com perdas**				**Sem perdas**	
	Nosso ***(adaptativo ao conteúdo)***	(SONG, ZHOU, *et al.*, 2010)	(MA e SEGALL, 2011)	(GUPTE, AMRUTUR, *et al.*, 2011)	**Nosso** ***(apenas caminho G3)***	(SILVEIRA, GRELLERT, *et al.*, 2012)
Alvo	**MVC**	AVC	AVC	AVC	**MVC**	AVC
Adaptativo ao Conteúdo?	**Sim**	Sim	Não	Não	**Não**	Não
Ganhos de Energia	**69.5%**	25-50%	21-31%	17-24%	**51.3%**	24%
BD-PSNR	**-0,01 dB**	-0,04 dB	*N.I.*	-0,01dB	**0 dB**	0 dB
BD-BR	**0,18%**	1,36-3,92%	0,38-21%	0.7%	**0%**	0%

A Tabela C.1 mostra que a nossa técnica de compressão adaptativa ao conteúdo é capaz de reduzir a propagação de erro de modo a atingir perdas insignificantes nos indicadores de BD-PSNR e BD-BR. A nossa adaptação ao conteúdo, entretanto, supera a estratégia adaptativa (SONG, ZHOU, *et al.*, 2010) em todos os aspectos: redução do número de acessos à memória externa (39% de ganho) e eficiência taxa-distorção (BD-PSNR 0.03dB maior). Uma linha adicional na Tabela C.1 foi inserida para a comparação com o trabalho (SILVEIRA, GRELLERT, *et al.*, 2012) com uma versão não adaptativa do nosso caminho de compressão, o qual utiliza apena a tabela Huffman e a quantização referente ao caminho G3. Nessa comparação, a nossa estratégia ainda assim apresenta melhores resultados, superando o trabalho relacionado em 27,3% na redução de acessos à memória externa.

C.5 Conclusões e Trabalhos Futuros

Uma hierarquia de memória para os módulos de Estimação de Movimento e de Disparidade da Codificação de Vídeo Multivistas foi proposta neste trabalho. A base da estratégia proposta é o esquema de reuso de dados centrado nas referências da codificação. A memória interna foi desenvolvida juntamente com técnicas de power-gating e união dos blocos candidatos, visando reduzir a energia consumida por esta memória interna. A nossa arquitetura de memória provê até 71% de redução no consumo de energia da memória externa, quando comparado com a estratégia MBDR tradicional. A energia ligada à memória interna é reduzida em 85% e 65% para os componentes estático e dinâmico, respectivamente. A compressão de quadro de referência adaptativa ao conteúdo provê uma redução de 69.5% dos acessos à memória externa, o melhor resultado quando comparado com os trabalhos relacionados.

C.6 Referências

CHEN, C.-Y. et al. Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 16, n. 4, p. 553-558, Abril 2006.

GUPTE, A. D. et al. Memory Bandwidth and Power Reduction Using Lossy Reference Frame Compression in Video Encoding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 21, n. 2, p. 225-230, February 2011.

HUFFMAN, D. A. A method for the Construction of Minimum-redundancy Codes. **IRE**, v. 40, n. 9, p. 1098-1101, 1952.

JVT TEAM. **Common Test Conditions for Multiview Video Coding**. Doc. JVT-T207. [S.l.]: [s.n.]. 2006.

JVT TEAM. **Editors' draft revision to ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding – in preparation for ITU-T SG 16 AAP Consent (in integrated form)**. Doc. JVT-AA07. [S.l.]: [s.n.]. 2009.

MA, Z.; SEGALL, A. Frame Buffer Compression for Low-Power Video Coding. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, Brussels, BEL, 2011. **Proceedings…** IEEE: New York, USA, p. 757-760.

MAX, J. Quantizing for minimum distortion. **IRE Transactions on Information Theory**, v. 6, n. 1, p. 7-12, March 1960.

MERKLE, P. et al. Efficient Prediction Structures for Multiview Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, Piscataway, v. 17, n. 10, p. 14611473, November 2007.

MICRON. **1Gb: x16, x32 Mobile LPDDR SDRAM**. [S.l.], p. 95. 2007.

SHAFIQUE, M. et al. Adaptive Power Management of On-Chip Video Memory for Multiview Video Coding. In: ACM/IEEE/EDA DESIGN AUTOMATION CONFERENCE, San Francisco, USA, 2012. **Proceedings…** IEEE: New York, USA. p. 866-875.

SILVEIRA, D. et al. Memory bandwidth reduction in video coding systems through context adaptive lossless reference frame compression. In: SOUTHERN PROGRAMMABLE LOGIC CONFERENCE, Bento Gonçalves, BRA, 2012. **Proceedings…** IEEE: New York, p. 1-6.

SINGH, H. et al. Enhanced leakage reduction techniques using intermediate strengs power gating. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 15, n. 11, p. 1215-1224, November 2007.

SONG, L. et al. An adaptive bandwidth reduction scheme for video coding. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS. Paris, FRA, 2010. **Proceedings…** IEEE: New York, USA, p. 401-404.

TSUNG, P.-K. et al. System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint. In: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, New Orleans, USA, 2007. **Proceedings…** IEEE: New York, USA, p. 1001-1004.

WANG, Z. et al. Memory efficient lossless compression of image sequences with JPEG-LS and temporal prediction. In: PICTURE CODING SYMPOSIUM, Krakov, POL, 2012. **Proceedings…** IEEE: New York, USA, p. 305-3008.

XIU-LI, T.; SHENG-KUI, D.; CAN-HUI, C. An analysis of TZSearch algorithm in JMVC. In: INTERNATIONAL CONFERENCE ON GREEN CIRCUITS AND SYSTEMS, Shangai, CHI, 2010. **Proceedings…** IEEE: New York, USA, p. 516-520.

ZATT, B. et al. A low-power memory architecture with application-aware power management for motion & disparity estimation in Multiview Video Coding. In:

IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, San Jose, USA, 2011. **Proceedings…** IEEE: New York, USA, p. 40-47.

ZATT, B. et al. Run-Time Adaptive Energy-Aware Motion and Disparity Estimation in Multiview Video Coding. In: DESIGN AND AUTOMATION CONFERENCE, San Diego, USA, 2011. **Proceedings…** IEEE: New York, USA, p. 1026-1031.

ANNEX – PAPERS PUBLISHED DURING THE MASTER DEGREE

Paper #1:

SAMPAIO, F.; ZATT, B.; SHAFIQUE, M.; AGOSTINI, L.; HENKEL, J.; BAMPI, S. "Energy-Efficient Memory Hierarchy for Motion and Disparity Estimation in Multiview Video Coding." In: Design, Automation & Test on Europe (DATE 2013). (accepted for publication).

Paper #2:

SAMPAIO, F.; ZATT, B.; SHAFIQUE, M.; AGOSTINI, L.; HENKEL, J.; BAMPI, S. "Content-Adaptive Reference Frame Compressor Based on Intra-Frame Prediction for Multiview Video Coding". In: IEEE International Conference on Image Processing (ICIP 2013). (submitted).

Paper #3:

ZATT, B.; SAMPAIO, F.; SHAFIQUE, M.; AGOSTINI, L.; BAMPI, S.; HENKEL, J. "Run-Time Adaptive Energy-Aware Motion and Disparity Estimation in Multiview Video Coding." In: Design and Automation Conference (DAC 2011). 2011, pp. 1026-1031. (*Received a 'European Network of Excellence on High Performance and Embedded Architecture and Compilation' (HiPEAC) Paper Award*).

Paper #4:

SAMPAIO, F.; ZATT, B.; AGOSTINI, L.; BAMPI, S. "Memory Efficient FPGA Implementation of Motion and Disparity Estimation for the Multiview Video Coding." In: Southern Programmable Logic Conference (SPL 2012). 2012, pp. 1-6.

Energy-Efficient Memory Hierarchy for Motion and Disparity Estimation in Multiview Video Coding

Felipe Sampaio[2], Bruno Zatt[1,2], Muhammad Shafique[1], Luciano Agostini[3], Sergio Bampi[2], Jörg Henkel[1]
[1]Karlsruhe Institute of Technology (KIT), Chair for Embedded Systems, Karlsruhe, Germany
[2]Federal University of Rio Grande do Sul (UFRGS), Informatics Institute, PPGC-PGMICRO, Porto Alegre, Brazil
[3]Federal University of Pelotas (UFPel), GACI, Pelotas, Brazil
{bzatt, bampi, felipe.sampaio}@inf.ufrgs.br, agostini@inf.ufpel.edu.br, {muhammad.shafique, henkel}@kit.edu

*Abstract***— This work presents an energy-efficient memory hierarchy for Motion and Disparity Estimation on Multiview Video Coding employing a Reference Frames-Centered Data Reuse (RCDR) scheme. In RCDR the reference search window becomes the center of the motion/disparity estimation processing flow and calls for processing all blocks requesting its data. By doing so, RCDR avoids multiple search window retransmissions leading to reduced number of external memory accesses, thus memory energy reduction. To deal with out-of-order processing and further reduce external memory traffic, a statistics-based partial results compressor is developed. The on-chip video memory energy is reduced by employing a statistical power gating scheme and candidate blocks reordering. Experimental results show that our reference-centered memory hierarchy outperforms the state-of-the-art [7][13] by providing reduction of up to 71% for external memory energy, 88% on-chip memory static energy, and 65% on-chip memory dynamic energy.**

Index Terms **— Multiview Video Coding, MVC, 3D-Video, Low-Power Design, On-Chip Video Memory, Application-Aware DPM, Memory Hierarchy, Energy Efficiency, Motion Estimation, Disparity Estimation.**

I. Introduction

Increasing demands for immersive multimedia systems have driven the popularization of the 3D-video technology that embraces a wide range of applications such as cinema, automotive, telepresence, 3D (mobile) camcorders, etc. 3D videos are based on the multiview concept [1] where multiple independent cameras record the same 3D scene from different viewpoints. Multiple video streams represent huge amount of data that must be processed and encoded before storage or transmission. The Multiview Video Coding (MVC) standard [2] provides 20-50% increased coding efficiency in comparison to H.264/AVC [3]. This is due to the inter-view prediction using Disparity Estimation (DE), which results in significant increase in the encoding complexity and energy consumption. Along with the Motion Estimation (ME), the DE represents about 90% of the encoder energy consumption [13]. Therefore, ME/DE is the main optimization focus for energy reduction in the MVC encoders.

ME/DE is used to search an image region (candidate block) that presents the best matching in the reference frame (previously decoded frames). The search is performed within a *search window* using a search algorithm like TZ Search [16]. This search window is typically fetched from external memory and stored in an on-chip video memory. Even for fast search algorithms, frequent memory accesses and large on-chip memory requirements lead to high energy consumption. Moreover, since the memory energy contributes to approximately 90% of the total ME/DE energy consumption [13], on/off-chip memory energy reduction is mandatory to meet the constraints and design requirements posed by the mobile battery-powered devices.

Recent works have proposed solutions to reduce the number of external memory accesses by employing current macroblock (MB)-centered data reuse (MBDR) schemes [6][7]. However, candidate-based and search window-based MBDR data reuses suffer from increased external memory traffic and on-chip memory, respectively (mainly for MVC that demands multiple references and at least a 193x193 search window [11]). Looking forward to jointly address on/off-chip memory energy issues, asymmetric search window [11], search window follower [12], and dynamic search window formation [13] schemes were proposed. Asymmetric search window, however, might lead to undesirable quality losses in case of vertical motion/disparity scenarios while dynamic search window leads to irregular memory access and on-chip memory misses. A novel perspective to face ME/DE memory issues was proposed in [9] and [8]. In these solutions a reference frame region is fetched and multiple MBs are processed using the available data. It eliminates reference frame retransmissions at the cost of partial search results (motion/disparity vectors and similarity) storage due to out-of-order processing. This solution, however, does not properly consider the impact of partial results memory traffic and on-chip video memory size.

The main challenge is to *jointly minimize on-chip and off-chip energy consumption in order to reduce the overall energy related to ME/DE on MVC.*

Previous works also aimed to reduce the memory related energy consumption of ME/DE on MVC encoders. In [13], we presented a low power ME/DE architecture that uses the concept of search map and dynamic search window formation. The work [15] extends the previous work by employing a multi-sleep state model on-chip memory to reduce the leakage energy. Latest, in [14] we proposed an on-chip memory power management based on such techniques, like the search direction elimination, to reduce even more the on-chip memory energy. Now, in this work we proposed jointly on-chip and off-chip memory energy savings techniques. Besides, different from previous work, we proposed a regular off-chip memory access pattern to exploit the burst reads (regarding to the DDR memories) to be more energy-efficient than the irregular-pattern previous solutions.

A. Our Novel Contributions

This work proposes a novel reference-centered memory hierarchy for ME/DE on MVC targeting low-energy consumption at both on-chip storage and off-chip memory access. The memory hierarchy is composed of an on-chip video memory, an on-chip memory power gating control, and a partial search results compressor. Additionally, a customized search control unit is proposed to exploit the search behavior to achieve further energy reduction.

978-3-9815370-0-0/DATE13/©2013 EDAA

Our novel contributions are:

- **Reference-Centered Memory Hierarchy** that employs a Reference-Centered Data Reuse (RCDR) scheme. It makes the reference frames the center of processing order to avoid search window retransmission and to eliminate the need to simultaneously store on-chip multiple search windows. A novel memory access scheduling and an energy-efficient on-chip memory organization are proposed.
- **Statistics-Based Partial Results Compressor**: The out-of-order processing inherent to RDCR imposes partial results (motion/disparity vectors and SAD) storage. Statistically defined non-uniform quantization and Huffman coding are employed for partial results compression.
- **On-Chip Video Memory**: The on-chip memory is organized in multiple SRAM banks featuring line-level power gating capability. At run-time, search window regions that are less likely to be used are power-gated. Additionally, the candidate blocks coding order is rearranged to minimize on-chip memory line switching and, consequently, the dynamic energy consumption.

Paper Organization: Section II presents a motivational ME/DE memory case study. Section III describes the proposed memory hierarchy and data reuse scheme. In Section IV, the experimental results and comparison to state-of-the-art are presented. Section V concludes the paper.

II. ME/DE Memory Analysis: A Case Study

In this section a ME/DE memory analysis is presented to motivate the use of Reference-Centered Data Reuse (RCDR). Fig. 1. shows the number of accesses for one ME/DE search on one given reference frame. It is possible to note that some pixels in the reference frame are accessed more than 1000 times. The scenario becomes more challenging for MVC because its prediction structure demands the search on multiple reference frames to achieve efficient compression; see Fig. 2.

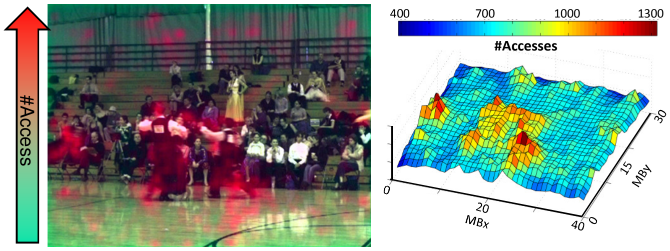


Fig. 1. Memory Access Analysis

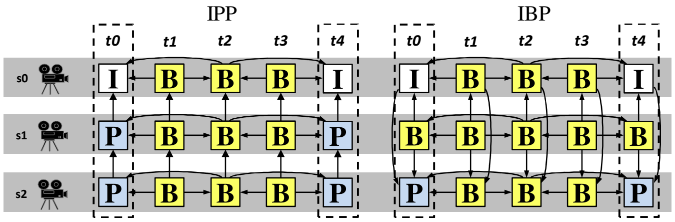


Fig. 2. MVC Prediction Structure, MB-Centered Perspective

Fig. 2. presents the MVC hierarchical prediction structure at the traditional current MB-centered perspective, where the origin of the arrows represent the current processed frame and the arrow head points to the requested reference frame. Examples for "IPP" and "IBP" view coding orders are provided. For simplicity, we use 3 views and Group of Picture (GOP, interval between I frames) equals 4; however, for real applications the GOP is typically higher and more views may be used [3]. Note that many frames are referenced multiple times to predict other frames. For instance, frame "S0T0" is referenced three and four times for "IPP" and "IBP" orders, respectively. It leads to a multiplication on the number of memory accesses (for MB-centered search) shown in Fig. 1. Moreover, as MVC requires multiple reference frames, they must be simultaneously stored on-chip. For instance, frame "S1T1" requires four references to perform the complete ME/DE.

Naturally, reference data is not fetched on demand from external memory every time it is required. Data reuse techniques like [6][7][11][12][13] reduce the external memory accesses by partially storing the data on-chip. In Fig. 3. the design spaces corner cases are presented for the on-chip vs. off-chip tradeoff considering 8 views, GOP=8, "IPB" and search window equals 193x193. In case no on-chip memory is employed, a huge external bandwidth is required. In contrast, if the all reference frames are stored on-chip, the on-chip memory grows drastically. Level-C [7] (a well known search window-based data reuse scheme) presents an intermediate compromise in this tradeoff but large on-chip memory and numerous external memory accesses are required due to MBDR limitation. To address this issue, our memory hierarchy employs a reference-centered data reuse scheme.

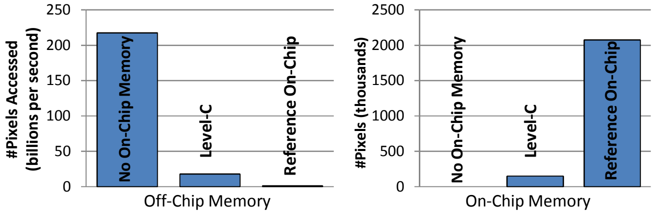


Fig. 3. Off-Chip vs. On-Chip Memory Tradeoff

III. Reference-Centered Memory Hierarchy

Fig. 4. presents the architecture of our video memory hierarchy employing reference-centered data reuse scheme. To control the ME/DE search and, consequently, the memory access pattern a Search Control unit is defined. Firstly, the Search Control sends search window requests to the memory. These requests are represented in video positions format. Therefore Address Generation Units (AGUs) are used to translate the requests to multiple external memory positions. Once the data is stored on-chip, a burst of candidate block positions is generated by the Search Control according to the TZ Search algorithm [16]. These candidate positions are rearranged by the energy-aware candidates merging unit in order to reduce the number of on-chip memory line switching. An on-chip memory power gating control monitors the search statistics and power-gates the on-chip memory lines accordingly. The candidates are processed by an array of processing elements (not described in this work). The best matching candidate and its SAD are forwarded to the search control. Due to the out-of-order processing inherent to RCDR, the temporary motion/disparity vectors and SAD (Sum of Absolute Differences) values must be stored for mode decision. These partial results are compressed using statistic-based non-uniform quantization and Huffman coding. As the partial results compressor employs variable-length coding, the partial results data is only sent to external memory once the local buffer is full. A specific AGU is implemented for partial results data.

Aware of the possible memory contention created by multiple AGUs requesting access to the external memory, we propose the fixed memory access scheduling presented in Fig. 5. Firstly, the Current MBs are fetched in order to allow the ME/DE processing start (part of the search window is already on-chip) followed by the missing search window column reading. In the following, if the partial results buffer is full, partial results are sent to external memory. Finally, the memory is available for other MVC blocks. Note that the number of current MBs (*D*) and search window blocks (*n*) vary at frame level changing the duration of the schedule intervals. Still, the schedule is not affected.

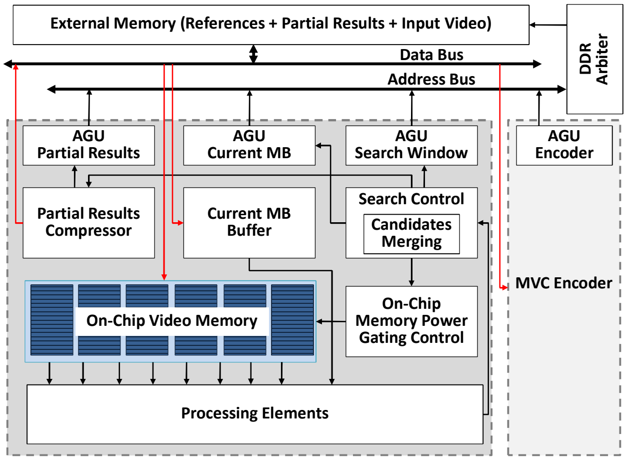


Fig. 4. Reference-Centered Memory Architecture

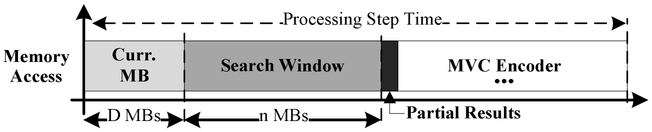


Fig. 5. External Memory Access Scheduling

A. Reference-Centered Data Reuse

The RCDR uses inverted dependence logic between reference frames and current MB. In this approach, the reference search window is fetched from external memory and those MBs requiring that specific data are processed. In other words, the reference data “calls” the MBs to be processed. For this reason, we define the term *dependent frames* for those frames “called” by a given reference frame.

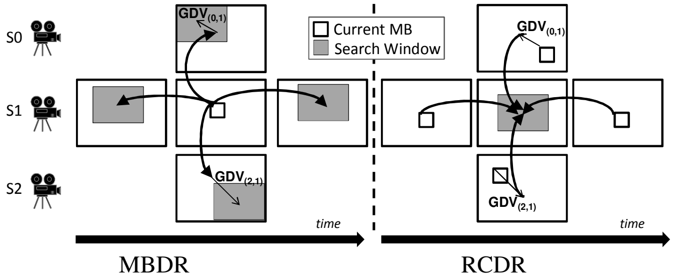


Fig. 6. MBDR vs. RCDR Data Reuses for ME/DE

Fig. 6. depicts the distinctions between search window-based MBDR and RCDR. Observe that in MBDR each current MB requests up to four search windows demanding 4x increased on-chip memory. Additionally, as those search windows are required more times in the future (to encode other frames), external memory data retransmission is needed. In contrast, on-chip storage of a single search window is required for RCDR resulting in reduced on-chip memory. Moreover, in RCDR the search window is requested and read from external memory a single time. Indeed, current MBs belonging to dependent frames are accessed multiple times. However, this represents a small impact for both on/off-chip memories, as demonstrated in results section. The GDV (Global Disparity Vector) is taken into consideration to locate the current MBs positions, as shown in Fig. 6.

Impact on External Memory Access: The equations presented below quantify the differences between MBDR and RCDR in terms of external memory accesses and on-chip storage. We consider square SWs with size multiple of 16 pixels (i.e., integer number of MBs). Eq. (1) and (2) represent the on-chip (*OC*) memory size, in number of MBs, for MBDR and RCDR, respectively. Where the search window size is *nxn*; *R* and *D* denote the number of reference or dependent frames, respectively. For MBDR, the quadratic factor related to search window size is further multiplied by the number of reference frames leading to a accentuated on-chip memory increase.

$$OC_{MBDR} = Rn^2 + 1 \quad (1)$$

$$OC_{RCDR} = n^2 + D \quad (2)$$

Eq. (3) and (4) show the external bandwidth (BW_{Step}), in number of MBs, for each *search step* after the on-chip memory is full. The search step is defined as one MB processing in MBDR case and a search window processing in case of RCDR. For each step MBDR reads *R* search window columns while RCDR reads one column plus *D* MBs. Note that MBDR suffers with the search window increase. The bandwidth for a frame line (BW_{Line}) is obtained applying Eq. (5) and (6). BW_{Line} has two components; the line initial read cost (between brackets) and the (*W-1*) steps red cost (curly brackets), where *W* denotes the number of MBs in a frame line. Note, that the *R* factor multiplies n^2 and *Wn* components in case of MBDR leading to strong bandwidth increase with the number of reference frames and search window size. To obtain the total bandwidth for a frame we multiply Eq. (5) and (6) by the number of MB is a frame column (*H*).

$$BW_{MBDR,Step} = Rn + 1 \quad (3)$$

$$BW_{RCDR,Step} = n + D \quad (4)$$

$$BW_{MBDR,Line} = [(Rn^2 + 1)] + \{(W - 1)(Rn + 1)\} \quad (5)$$

$$BW_{RCDR,Line} = [(n^2 + D)] + \{(W - 1)(n + D)\} \quad (6)$$

The third memory bandwidth component, omitted in Eq. (3)-Eq.(7), is the traffic spent to write partial results to the external memory. Although less representative in terms of amount of data, it accesses memory frequently leading to memory contention and efficiency loss. To address these issues, a partial results compressor with buffering is presented.

B. Statistics-Based Partial Results Compressor

The partial results are composed of two distinct data types, (i) motion/disparity vectors and (ii) SAD values, that present distinct numerical range and statistical behavior. For this reason, we discuss them separately. Fig. 7. a shows the histogram of the disparity vectors extracted from *Poznan Carpark* video sequence. Although the disparity vectors are mainly concentrated in DV_x=0, multiple disparity vectors are distributed in a wide value range complicating the compression step.

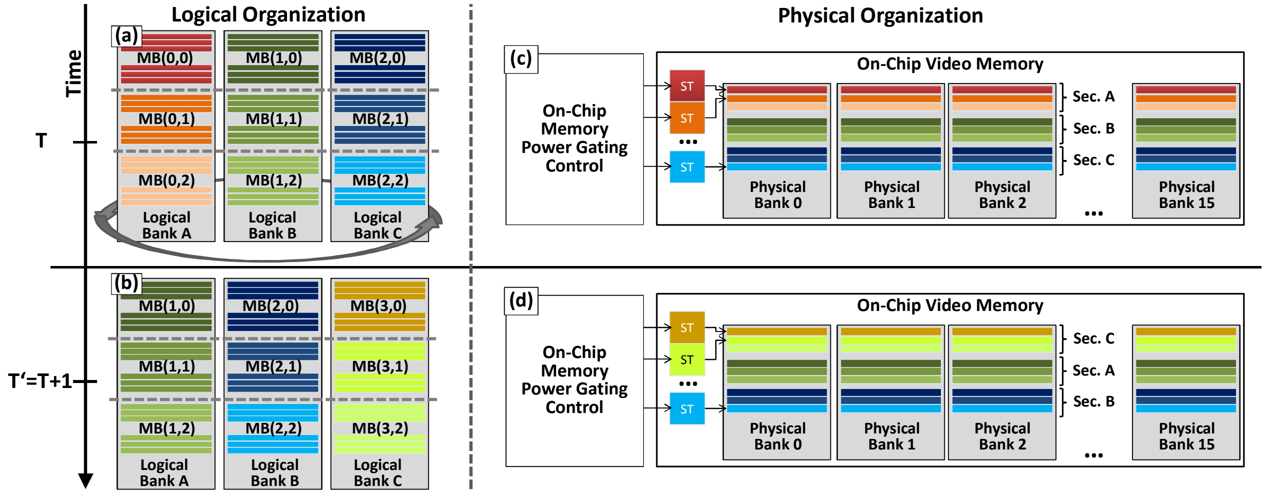


Fig 8. On-Chip Video Memory (a)(b) Logical and (c)(d) Physical Organization

To concentrate these vectors in a reduced range, distinct predictors were evaluated. A median spatial predictor (above and left MBs) provided the best results, as shown by the histogram and PDFs (Probability Density Function) in Fig. 7. b. The differential disparity vectors distribution is concentrated in a small range around $DV_x=0$ (see Fig. 7. b). For 54 values that represent $\mu\pm2\sigma=95.8\%$ of differential vectors occurrences, a Huffman table was generated according to the techniques presented in [17]. The differential vectors values out of this range are represented by a special Huffman value followed by the vector value in 8-bit binary representation. Analogous statistical analysis and specific Huffman table definition were performed for differential motion vectors.

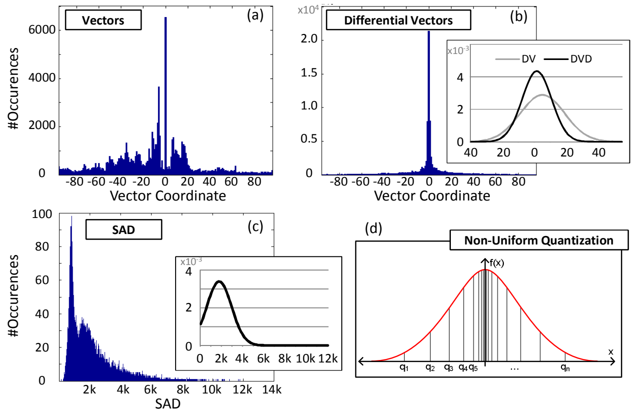


Fig. 7. Partial Results Compressor Statistics

In Fig. 7. c the histogram and PDF of SAD values for DE are presented. As can be noticed, SAD values are spread along a wide numerical range. Still, high concentration is observed around 1k. For such distribution, quantization is required. To reduce the impact of quantization errors, we employ a non-uniform quantization designed for a Gaussian distribution according to Lloyd algorithm and a Lloyd-Max refinement [18]. The quantizer employs 512 levels optimized for minimum mean square error (MMSE). The non-uniform quantization employs a reduced quantization step for regions with high occurrence (close to the average) and larger quantization step in regions with less occurrences (PDF tails), as represented in Fig. 7. d. After quantization the quantized SADs are encoded using a 189-entries Huffman table. SAD values out of range are encoded by a special Huffman value followed by the SAD value in 14 bits. After that, the partial results are concatenated in a 512-bits local buffer. Once this buffer is full, it is written to the external memory following the schedule defined in Fig. 5. The partial results AGU generates serial addresses in a memory region specific for partial results.

RCDR and partial results compression contribute mainly to external memory-related energy reduction. Below we present strategies to reduce on-chip memory energy consumption.

C. On-Chip Video Memory Organization

Our on-chip video memory is logically defined as a circular buffer organized in a 2D-array fashion to provide direct matching to video data. It is composed by *B* (where *B*=*n*) logical memory banks that rotate after each *search step* to avoid retransmission of overlapping SW. Fig. 8a shows a simplified example with 3x3-MBs search window (each MB is represented by a distinct color) in time instant *T*. The rotation for time instant *T'=T+1* (after a *search step*) is presented in Fig. 8b where the leftmost column of MBs (MB(0,x)) is dropped and a new column at the right is fetched (MB(3,x)). Columns MB(1,x) and MB(2,x) are reused. This organization, however, is not suitable for physical implementation once ME/DE requires MB parallel read.

The physical organization of our on-chip video memory is presented in Fig. 8c and Fig. 8d for time instants *T* and *T'=T+1*, respectively. It is composed of 16 parallel 128-bits wide SRAM banks to store 16 reference pixels per bank line. Each memory line stores and feed one complete MB in parallel. Each bank is further divided in sectors of *n* lines representing one search window column. The total number of lines is defined by the number of MBs in the search window (n^2). Note that differently from the logical organization, MBs columns are not shifted for every *search step*. For that the memory sectors are renamed accordingly, as depicted in Fig. 8. Line-level power gating is employed to support fine-grained power management; the power gating management is discussed in next section.

D. On-Chip Video Memory Power Gating

We propose a statistical power gating scheme that employs multiple SRAM sleep modes in order to reduce the static energy consumption due to the leakage current. Differently from related work solutions [13], this scheme does not require image properties extraction or MB-level memory access prediction in order to provide a light-weight (but still efficient) solution. Four power states are implemented [19]: S0=OFF (Vdd=0), S1= Data Retentive (Vdd=Vdd*0.3), S2= Data Retentive (Vdd=Vdd*0.5) and S3= ON (Vdd= Vdd). Where each state has an associated wakeup energy cost ($WE_{S0}> WE_{S1}> WE_{S2}> WE_{S3}=0$). For this reason, regions that are frequently accessed are mapped to *S3*, unused regions to *S0*, and other regions are mapped to *S1-S2* according to run-time statistics.

```
1.  onChipPowerGating(n, v, CurrFrame, offStatMap)
2.  D {ME, DE} ← getNumberDependentFrames(v, CurrFrame);
3.  PowerMap_SW ← So;                    // PowerMap initialization
4.  For all MB ∈ nxn                     // for all MBs in the nxn SW
5.      If (FirstFrame)   // if first frame
6.      Then                  // use offline statistics
7.         StatMap_SW = D {ME}*offStatMap_ME + D {DE}*offStatMap_DE;
8.      Else                  // use statistics from previous encoded frames
9.         onStatMap ← getPrevFramesStat(v, CurrFrame);
10.        StatMap_SW = D {ME}*offStatMap_ME + D{DE}*offStatMap_DE;
11.     EndIf
12.     $PowerMap_{SW} = \begin{cases} S1 & If \quad \mu - 2\sigma < StatMap_{SW}(x,y) < \mu - 3\sigma \\ S2 & If \quad \mu - \sigma < StatMap_{SW}(x,y) < \mu - 2\sigma \\ S3 & If \quad Else \end{cases}$
13. End For
14. PowerMap_SW ← physicalMemPos(PowerMap_SW);
15. For all MB ∈ CurrFrame         // for all MBs in the frame
16.     PowerGate(PowerMap_SW );
17.     currStatMap ← performSearch();
18. End For
19. StoreCurrMap(v, CurrFrame, currStatMap); return;
```

Fig. 9. Pseudo-Code of the On-Chip Video Memory Power Gating

Fig. 9. presents our power-gating algorithm which is activated when a new frame processing starts. Firstly, the number of dependent frames is calculated (line 2) and the $PowerMap_{SW}$ is reset. $PowerMap_{SW}$ has one entry for each MB in the search window and if the used search window is smaller than the physical memory, all MBs exceeding the search window are fixed in *S0*. For each MB in the *nxn* search window (line 4), a weighted statistics map ($StatMap_{SW}$) is defined; offline statistics are used in case this is the first processed frame (line 7), otherwise statistics from the previously encoded frames are used (line 9-10). The weighting factors depend on the number of ME/DE (*D{ME,DE}*) dependent frames. It is required due to distinct memory access behavior between ME and DE. The $StatMap_{SW}$ is then converted to $PowerMap_{SW}$ by using statistically defined thresholds (line 12). The thresholds are calculated based on the memory access statistics (average and standard deviation) of each block within the search window. The $PowerMap_{SW}$ is finally mapped to the actual physical memory positions (line 14). For each MB in the frame the power gating signals are sent to the on-chip memory (line 16) and the ME/DE search is performed (line 17). At the end statistics are updated for further frames processing (line 19).

E. Energy-Aware Candidate Blocks Merging

Although static energy is becoming dominant in submicron on-chip memories, dynamic energy reduction significantly contributes to overall energy [20]. To avoid frequent on-chip memory line switching, we define an energy-aware candidate blocks merging strategy. As far as multiple dependent MBs are searching simultaneously in the same search window, multiple search points are requested multiple times. The abstract example using TZ search depicted in Fig. 10. shows the access pattern for MB A (left) and MB B (right). Note that for the first search step (dark gray blocks) all candidates are the same. Additionally, some candidates in the second step (black blocks) are repeated. In the figure center, bright blocks represent candidate blocks accessed by both MBs and dark blocks MBs accessed by a single current MB.

Our candidate blocks merger receives all search points generated by the search control and rearranges them in order to process together repeated candidates and avoid unnecessary SRAM line switching (address line switching, bitline pre-charge, sense amplifiers switching, output buffer switching). Moreover, the new processing order follows the left-right and up-down fashion so the processing can start even before the rightmost column is updated for each search step (Section C).

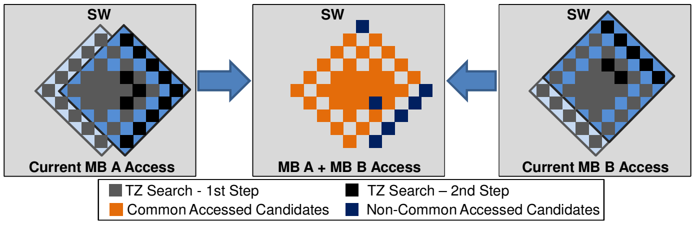


Fig. 10. **Example**: Candidate Blocks Merging

IV. Experimental Results

The experimental results were generated using real video sequences and coding settings recommended by JVT [4] running on the MVC reference software [5]. A customized energy simulator was used to measure the energy consumption of our approach and related solutions. The SRAM leakage and wake-up energies values were calculated based on [19]. The off-chip memory energy savings were evaluated by using the MT46H64M16LF LPDDR 1 Gigabit memory [21]. Note, the experimental results include the wakeup energy overhead.

Four video sequences including three video resolutions were used: *Ballroom* and *Flamenco2* (VGA-640x480), *Balloons* (XGA-1024x768), and *Poznan Carpark* (HD1080-1920x1920). The experiments included 4-views and 8-views sequences considering “IPP” and “IBP” view coding orders. Other settings are: CABAC, FRExt, QP={22,27,32,37}, GOP=8, TZ Search [2]. Note that among the algorithms proposed only SAD quantization may insert coding losses. However, due to the non-uniform quantization, no losses were observed in our experiments. Thus, coding results are omitted.

A. External Memory Energy Savings

Fig. 11. presents the off-chip energy savings for changing search window size and video resolution, under four distinct scenarios, compared to Level-C [7]. Observe that the energy savings scale well with the increase in number of views and search window. Additionally, our solution does not suffer with frame resolution increase. Higher savings happen in case of “IBP” due to more intense search window reuse, i.e., each search window is used by an increased number of current MBs. These results include energy reduction due to the partial results compression. Our compressor leads to 53.2% (average) external energy reduction for partial results communication (compared to the non-compression scenario), as detailed in Fig. 12. a.

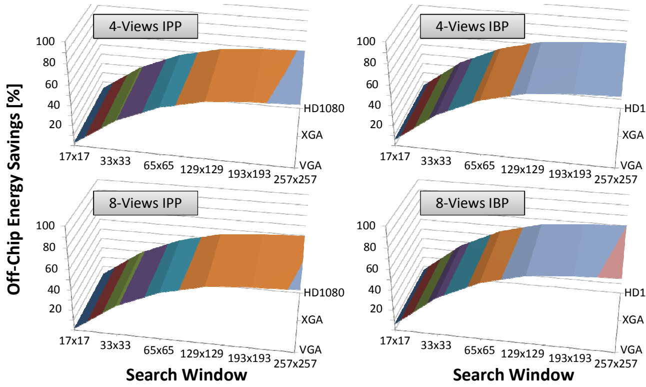


Fig. 11. Off-Chip Memory Energy Savings

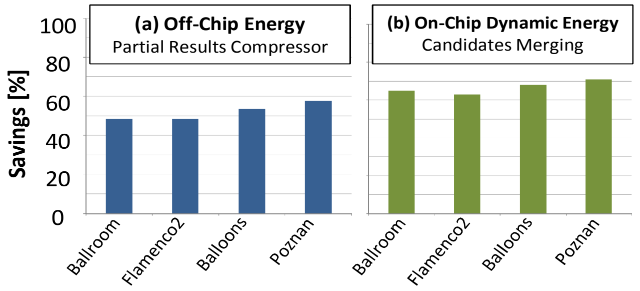


Fig. 12. Energy Savings due to (a) Partial Results Compressor and (b) Candidates Merging.

B. On-Chip Video Memory Energy Savings

Compared to the Level-C [7], our on-chip video memory size is significantly reduced (see Fig. 13.) because there is no need to simultaneously store multiple search windows on chip. Note that our on-chip memory grows smoothly with the search window increase. Moreover, compared to the search window storage, the cost (considered in Fig. 13.) for storing current MBs (that may reach 9 MBs for 8-views "IBP") is negligible. This cost is amortized as the search window increases. The reduced on-chip memory size directly leads to less static energy consumption, as shown in Fig. 14. Compared to [7], 77% energy reduction is reached without employing our power-gating technique. If the power-gating is used, further 88% of reduction is reached outperforming [13] in 61 %. These results refer to 4-views "IBP" scenario.

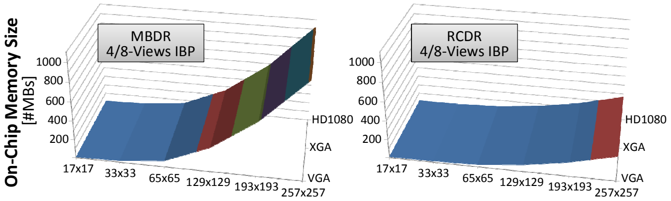


Fig. 13. On-Chip Memory Size Reduction

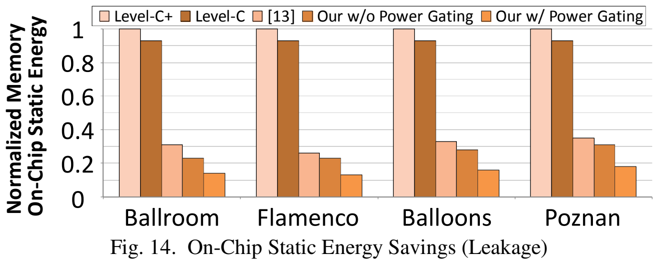


Fig. 14. On-Chip Static Energy Savings (Leakage)

In terms of on-chip dynamic energy, our candidate merging strategy reduces the energy consumption in 65%, as shown in Fig. 12. b. At the best of authors' knowledge, this is the first application specific technique to address the dynamic on-chip memory energy consumption for the ME/DE.

V. Conclusion

A memory hierarchy for Motion and Disparity Estimation on Multiview Video Coding was presented. It exploits a reference-centered data reuse scheme along with partial results compression and memory access scheduling in order to reduce external memory energy. An on-chip video memory organization with line-level power gating and candidates merging scheme is presented targeting on-chip energy reduction. Our memory architecture provides up to 71% off-chip memory energy reduction. On-chip memory-related energy is reduced on 88% and 65% for static and dynamic energies, respectively.

Acknowledgement

This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre "Invasive Computing" (SFB/TR 89); http://invasic.de.

References

[1] A. Smolic, et al. "Coding Algorithms for 3DTV - A Survey." In: IEEE TCSVT, v. 17, n. 11, pp. 1606-1621, nov. 2007.

[2] Joint Draft 8.0 on Multiview Video Coding, JVT-AB204, 2008.

[3] P. Merkle, et al. "Efficient Prediction Structures for Multiview Video Coding." In: IEEE TCSVT, v. 17, n. 11, pp. 1461-1473, nov. 2007.

[4] JVT. "Com. Test Cond. for Multiview Video Coding". JVT-T207, 2007.

[5] JMVC Reference Software , Sep. 2009.

[6] J.-C. Tuan, et al. "On the Data Reuse and Memory Bandwidth Analysis for Full-Search Block-Matching VLSI Architecture." In: IEEE TCSVT, v. 12, n. 1, p. 61-72, jan. 2002.

[7] C.-Y. Chen, et al. "Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders." In: TCSVT, v. 16, n. 4, p. 553-558, april. 2006.

[8] P.-K. Tsung, et al. "System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint". IEEE ISCAS p. 1001-1004, 2007.

[9] T.-C. Chen, et al, "Single Reference Frame Multiple Current Macroblocks Scheme for Multi-Frame Motion Estimation in H.264/AVC", In IEEE ISCAS, 2005, pp. 1790 – 1793.

[10] T.-Y. Kuo, et al. "A novel method for global disparity vector estimation in multiview video coding". In: IEEE ISCAS 2009.

[11] X. Xu, Y. He , "Fast disparity motion estimation in MVC based on range prediction," IEE ICIP, pp.2000-2003, 2008.

[12] S. Saponara, L. Fanucci, "Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems", IEE Comp. & Digital Tech., vol.151, no.1, pp. 51- 59, 2004.

[13] B. Zatt, M. Shafique, F. Sampaio, L. Agostini, S. Bampi, J. Henkel, "Run-time adaptive energy-aware motion and disparity estimation in multiview video coding", IEEE DAC, pp. 1026-1031, 2011.

[14] M. Shafique, B. Zatt, F. L. Walter, S. Bampi, J. Henkel, "Adaptive Power Management of On-Chip Video Mamory for Multiview Video Coding", IEEE DAC, pp. 866-875, 2012.

[15] B. Zatt, M. Shafique, S. Bampi, J. Henkel, "A Low-Power Memory Architecture with Application-Aware Power Management for Motion & Disparity Estimation in Multiview Video Coding", IEEE ICCAD, pp. 40-47, 2011.

[16] J. Yang et al., "Multiview video coding based on rectified epipolar lines", International CICSP, pp. 1-5, 2009.

[17] Huffman, D.A., "A Method for the Construction of Minimum-Redundancy Codes," IRE, vol.40, no.9, pp.1098-1101, Sept. 1952.

[18] Max, J.; , "Quantizing for minimum distortion," Information Theory, IRE Transactions on , vol.6, no.1, pp.7-12, March 1960.

[19] H. Singh et al., "Enhanced leakage reduction techniques using intermediate strength power gating", IEEE Transaction on Very Large Scale Integration, vol. 15, no. 11, pp. 1215-1224, 2007.

[20] S.l Rodriguez, B. Jacob,"Energy/power breakdown of pipelined nanometer caches (90nm/65nm/45nm/32nm", ISLPED, pp. 25-30, 2006.

[21] Micron. "1Gb: x16, x32 Mobile LPDDR SDRAM". Available at: www.micron.com. Last Accessed: December, 2012.

CONTENT-ADAPTIVE REFERENCE FRAME COMPRESSION BASED ON INTRA-FRAME PREDICTION FOR MULTIVIEW VIDEO CODING

Felipe Sampaio[1], Bruno Zatt[1], Muhammad Shafique[2], Luciano Agostini[3], Jörg Henkel[2], Sergio Bampi[1]

[1] Informatics Institute (PPGC-PGMICRO), Federal University of Rio Grande do Sul - Porto Alegre, Brazil
[2] Chair for Embedded Systems (CES), Karlsruhe Institute of Technology - Karlsruhe, Germany
[3]Group of Architectures and Integrated Circuits (GACI), Federal University of Pelotas - Pelotas, Brazil
{felipe.sampaio, bzatt, bampi}@inf.ufrgs.br, agostini@inf.ufpel.edu.br, {muhammad.shafique, henkel}@kit.edu

ABSTRACT

This paper presents a content-adaptive reference frame compression scheme to alleviate the large overhead of external memory communication during the Motion and Disparity Estimation process in Multiview Video Coding (MVC). Our scheme is based on a *simplified* intra-prediction process to reduce the spatial redundancy of the reference samples. The intra-prediction residue is compressed by a path composed of non-linear quantization and Huffman-based entropy encoder. Four different quantization strengths and Huffman tables were statistically defined. They are dynamically selected according to a content adaptation strategy, which classifies the original blocks based on their spatial homogeneity. Experimental results show that the proposed content-adaptive compression scheme is able to reduce the external memory accesses by up to 63% along with negligible losses in the MVC encoder rate-distortion performance. Compared to the best available related work [12] our content-adaptive reference frame compression achieves 39% reduced external memory accesses, while still providing a BD-PSNR increase of 0.03dB.

Index Terms— Reference frame compression, content adaptation, memory reduction, multiview video coding.

1. INTRODUCTION AND MOTIVATION

The increasing interest in immersive 3D multimedia applications has led to research and development focusing on mobile devices capable of 3D-video personal recording and displaying [1]. Due to the battery-powered nature of such devices, several energy constraints emerge when multiview videos (basis for the 3D perception) are processed. The state-of-the-art Multiview Video Coding (MVC) [2] standard provides 20%-50% increased coding efficiency in comparison to the H.264/AVC [3]. Besides new syntax elements to support multiview video representation, the key coding tool in MVC is the inter-view prediction, which uses the Disparity Estimation (DE) search to capture the objects displacement due to different camera positions. Along with the Motion Estimation (ME), the DE represents the most energy consuming module in the MVC encoder (more than 90% of the overall energy) [4][5][6].

Conceptually, the ME/DE goal is to search for an image region (candidate block) that presents the best matching in the reference frame (previously coded frames). The search is performed within a search window, which is accessed from the Decoded Picture Buffer (DPB); in real-world systems, the DPB is typically mapped to external DDR memories and locally stored in an on-chip video memory. According to work of [4][5][6], memory-related energy consumption represents 90% of total ME/DE energy consumption. This is primarily due to the extensive off-chip and on-chip memory accesses. **Therefore, memory access reduction is needed for energy-efficient and/or performance-efficient MVC encoding**.

To reduce the total number of external memory accesses, multiple data reuse schemes were proposed targeting both current Macroblock order [7][8] and reference data order [9][10]. However, these solutions lack efficiency in handling the increasing external memory demands imposed by the ME/DE in real-time MVC encoding. Note that for high definition HD1080p (1920x1080) MVC encoding, the current literature recommends the use of at least 193x193 search window size [3] for each reference frame. Fig. 1 quantifies this trend by presenting the ME/DE external memory bandwidth for MVC encoding at 30 fps considering two solutions: (i) ME/DE with no data reuse and (ii) ME/DE with Level-C [8] data reuse scheme (widely used search window-based data reuse scheme). Besides, the maximum transfer rates of three Low-Power DDR (LPDDR) families are plotted to draw a relation between MVC requirements and the latest embedded memory technologies [11].

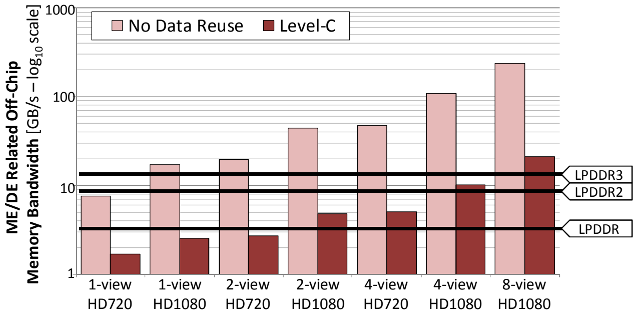


Fig. 1. ME/DE memory requirements

It can be noted that LPDDR memories are not able to meet the ME/DE external memory demands without data reuse techniques. Moreover, even employing efficient data reuse techniques, like Level-C (see the dark bar in Fig. 1), the current LPDDR3 technology cannot meet the demanded transfer rate for 8-views ME/DE. Note, in real systems the external memory access typically has to be shared with other applications and sustaining peak DDR performance is an unrealistic assumption. Therefore, *it is crucial for energy-efficient MVC-based encoding applications to use techniques for minimizing the external memory accesses during the ME/DE process*.

Recent works [12],[13],[14],[15] and [16] have taken a first step to address this problem by compressing the reference samples before storing them in the DPB external memory. In this case, the ME/DE must recover the original samples (by decompression) to use them as reference in the block matching process. Lossless DPB compression techniques were proposed in [16] and [15] using the JPEG-LS and the RFCAVLC, respectively. Moreover, lossy solutions based on quantization were proposed in [13] and [14]. The work [13] decomposes the full-resolution image into two components (low and high resolutions) and applies different compression path to each one. In [14], the MMSQ-EC is proposed as a low-complexity scheme based on the min-max scalar quantization technique. Video content characteristics are firstly exploited in [12], where a compression mode decision is proposed to find the best possible way to compress each region of the frame. However, these schemes have not been designed for the MVC memory constraints, which require much more ME/DE memory accesses. Besides, as the MVC prediction structure has relatively more dependencies than single-view encoders, errors are propagated along the GOP (Group of Pictures) not only along the temporal neighboring frames, but also for the neighboring views. Therefore, the main challenge is to *compress the reference data while minimizing the error propagation along the MVC prediction structure*.

Our Novel Contributions: This work proposes a novel content-adaptive reference frame compression scheme based on the intra-prediction encoder defined by the MVC. As a result, our frame compression is able to exploit already existing encoding information to *avoid* additional complexity. It also incorporates video content-driven adaptation capabilities that improve the DPB compression efficiency. Our scheme employs:

- **Intra Prediction-Based Frame Compression:** A simplified intra-prediction is performed to reduce the spatial redundancies present in the reference frames. To avoid additional computational overhead, our scheme inherits the best 4x4 intra mode evaluated by the MVC mode decision during the encoding process.
- **Content Adaptation for Residue Compression:** Once the redundancies are eliminated, the residues are forwarded to a non-linear quantization and Huffman-based variable length entropy encoding. To minimize the quantization error, our scheme dynamically selects the quantization strength that fits best to the video content characteristics. The Huffman tables and quantization strengths are designed considering the statistical behavior of the intra-prediction residue.

Paper Organization: Section 2 presents our novel content-adaptive compression scheme. The content adaptation strategy along with the statistical design for quantization and the Huffman tables is presented in Section 3. Section 4 presents the results for external memory access savings and comparison with state-of-the-art schemes. Section 5 concludes the paper.

2. CONTENT-ADAPTIVE DPB COMPRESSION SCHEME

The proposed scheme compresses the samples after they are completely encoded and reconstructed (i.e., after the Deblocking Filter). At this point, the reconstructed samples are stored in the DPB (external memory) that are later used as reference in the ME/DE of the subsequent frame(s). This operation flow is reflected in the Equations (1) and (2), where the compressed bandwidth of reading from and writing to (CBW_{read} and CBW_{write}) the external memory are expressed in terms of the DPB compression factor $\boldsymbol{\alpha}$.

$$CBW_{read} = \boldsymbol{\alpha}\, BW_{ME/DE} + \boldsymbol{\varepsilon_{read}} \quad (1)$$

$$CBW_{write} = \boldsymbol{\alpha}\, (W \times H) + \boldsymbol{\varepsilon_{write}} \quad (2)$$

The compression parameter $\boldsymbol{\alpha}$ reduces the ME/DE external memory accesses for fetching the reference samples ($BW_{ME/DE}$) and for writing the $W\times H$ reconstructed reference frame. W and H are the horizontal and vertical frame resolutions, respectively. The $\boldsymbol{\varepsilon}$ parameters represent the error transmission. The error resulting from the quantization step is stored, which is used at the decoder side for error compensation for avoiding mismatches between the references of the MVC encoder and decoder [14].

The pseudo-code of our content-adaptive algorithm scheme is presented in Fig. 2. The inputs are: (a) the reconstructed reference samples, (b) the available neighboring samples (thirteen at maximum), (c) the sixteen already-selected intra-prediction modes, and (d) the original samples of the current MB.

Content-Adaptive Reference Frame Compression Algorithm

```
1.  // inputs: origMB[]: original macroblock (MB) samples
2.  // reconMB[]: reconstructed MB samples
3.  // neighbors[]: thirteen neighboring samples for intra prediction
4.  // predModes[]: sixteen MVC encoder best intra prediction modes
5.  // outputs: codedMB[]: compressed bitstream to send to the DPB
6.  function compressMB (origMB[], reconMB[], neigh[], predModes[]):
7.  codedMB ← Ø
8.  foreach reconBlk4x4, origBlk4x4 in reconMB, origMB
9.        pred4x4 ← intraPrediction(neigh, predModes[blk4x4])
10.       resBlk4x4 ← pred4x4 – blk4x4
11.       hG ← HG(origBlk4x4)
12.       [quantedBlk4x4, errorsBlk4x4] ←quantize(resBlk4x4, hG)
13.       codedBlk4x4←huffmanEnco(quantedBlk4x4,huffTab[hG])
14.       codedMB.append(hG)
15.       codedMB.append(predModes[blk4x4])
16.       codedMB.append(codedBlk4x4)
17.       compensateErrors(errorsBlk4x4)
18. end loop
19. return codedMB
```

Fig. 2. Content-adaptive DPB compression algorithm

Our scheme is applied to every 4x4 block of a reconstructed Macroblock (*line 8*). Initially, a simplified intra-prediction using only 4x4 blocks is performed (*line 9*) to eliminate the spatial redundancies intrinsic to the reconstructed reference samples. In order to avoid additional computation, the proposed scheme inherits the best 4x4 intra mode calculated by the MVC mode decision. Note, our compressor uses the best 4x4 mode regardless of the mode selected for encoding the MB (that may be intra 16x16 or inter-frame/view). The simplified intra-prediction in our scheme is compliant to the H.264/AVC definition for 4x4 blocks: 9 possible modes using thirteen neighboring samples, when available [3]. Then, the residue (difference between the reconstructed and the predicted samples) is calculated (*line 10*). The residue values distribution is much more concentrated when compared to the reconstructed samples. Exploiting this concentrated distribution, the Huffman-based entropy encoder is applied [17]. The goal is to assign the smaller codes to the most likely symbols (*line 13*).

Since the intra-prediction exploits the spatial correlation of the image, the heterogeneous (textured) blocks tend to generate spreader distributions of values, which are not desirable for Huffman coding. To better deal with such blocks, the proposed scheme implements a non-linear quantization to further minimize the range of representation (*line 12*). By definition, the non-linear quantization applies smaller quantization steps (distance between two quantization levels) for the higher probability regions along the values statistical distribution (near the average), see Fig. 3d. Fig. 3a and Fig. 3b present the histogram of the raw reconstructed samples and the intra-coded residues, respectively, for the *flamenco2* test sequence. It can be noted that the spatial redundancy elimination (Fig. 3b) generates a concentrated zero-centered distribution, as also depicted in the Probability Density Function (PDF) of Fig. 3c. The quantization strength is defined by the number of levels that is employed, like in Fig. 3d. For example, 8-level non-uniform quantization means that the initial representation (considered as 8 bits) will be reduced from [-128,127] to [-4,3].

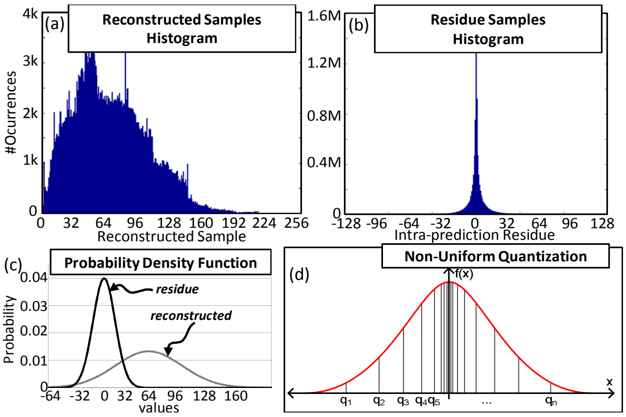


Fig. 3. (a) Reconstructed and (b) residue values histogram and (c) PDFs, and (d) non-linear quantization

The final step is to pack the compressed residues (Huffman output) and the prediction modes for all 4x4 blocks (3 bits for each mode) into fixed-sized packages to be sent to the external memory (*lines 15-16*). Besides, in order provide equal references in the MVC encoder and decoder, an additional external memory is required to save the quantization errors to be compensated during the decoding process [14].

After the quantization, the initial samples cannot be recovered identically due to the range discretization. It leads to MVC encoder drops on rate-distortion efficiency. Regarding the high number of dependencies on the prediction structures (temporal and disparity domains), these errors may propagate along all ME/DE operations inside the GOP. To handle with this issue, we propose a content-adaptive strategy to adapt the Huffman table and the quantization step according to the image characteristic (*lines 11-14* of Fig. 2). Further aspects of this dynamic adaptation are described in the following.

3. CONTENT ADAPTATION AND HUFFMAN/QUANTIZATION DESIGN

The proposed content adaptation of Huffman-based entropy encoder and non-linear quantization steps aims at jointly minimizing the intra-predicted residue and the quantization errors. The residue minimization leads to external memory access savings, while minor quantization errors provide less degradability in the MVC encoding efficiency. Based on statistics from the original frame, the proposed technique classifies the 4x4 original blocks according to their homogeneity. The intra-prediction over homogeneous blocks provides concentrated residue distribution. Therefore, a small number of quantization levels is required. As a result, a homogeneous block leads to reduced error and efficient entropy encoding. In contrast, textured blocks should be quantized using weaker quantization in order to minimize the errors inserted in the ME/DE MVC encoding loop. It is observed that different regions within the same frame may exhibit diverse spatial image properties, as depicted in the variance map of Fig. 4a. Thus, a block-level content adaptation is needed in order to achieve high DPB compression with minimal quality losses.

Our content adaptation scheme classifies the 4x4 blocks in four homogeneity groups *HG=[G0,G1,G2,G3]* according to Equation (3), where homogeneity is measured using the block variance, $\boldsymbol{\sigma}^2$ in Equation (4). Note, the variance calculation is performed using the original blocks (*origBlk4x4*), thus avoiding the data dependencies within the encoder loop.

$$\boldsymbol{HG}(origBlk4x4) = \begin{cases} G0 \;\; \boldsymbol{if} \;\; 0 \leq \boldsymbol{\sigma}^2(origBlk4x4) < TH_0 \\ G1 \;\; \boldsymbol{if} \;\; TH_0 \leq \boldsymbol{\sigma}^2(origBlk4x4) < TH_1 \\ G2 \;\; \boldsymbol{if} \;\; TH_1 \leq \boldsymbol{\sigma}^2(origBlk4x4) < TH_2 \\ G3 \;\; \boldsymbol{if} \;\; TH_2 \leq \boldsymbol{\sigma}^2(origBlk4x4) \end{cases} \quad (3)$$

$$\boldsymbol{\sigma}^2(B) = \frac{1}{n^2}\sum_{j=0}^{n}\sum_{i=0}^{n}(B_{i,j} - \mu)^2 \quad (4)$$

Extensive simulations over a benchmark of videos were performed to statistically define thresholds (TH_0, TH_1 and TH_2), quantization intervals, and Huffman tables to better exploit the homogeneity of each residue group. Four different non-linear quantizations were defined: *nLev(G0)=8*, *nLev(G1)=16*, *nLev(G2)=32* and *nLev(G3)=256* (lossless), where *nLev* is the number of quantization intervals (levels). They were adapted to achieve the best possible efficiency (joint error and residue minimization) for the specific homogeneity property of each group. Allied to the quantization design, four different static Huffman tables were designed to have the best possible fit with the quantized coefficients of each group. The Huffman encoder is composed of: 8-entry table for *G0*, 16-entry table for *G1*, 32-entry table for *G2*, and 256-entry table for *G3*.

Fig. 4 presents an example for our content adaptation scheme for the *flamenco2* test sequence. The variance map depicted in Fig. 4 shows that the degree of homogeneity is exposed from the low variance (homogeneous regions) to the high values (textured regions). The residue classification, using Equation (3), can be directly performed over the variance map. Fig. 4b presents the blocks of the image classified in different block groups.

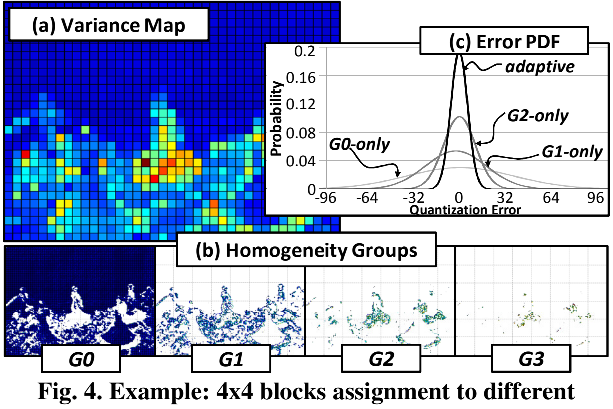


Fig. 4. Example: 4x4 blocks assignment to different quantization strengths (*flamenco2* test sequence)

The content adaptation provides relative minimal error propagation along the prediction structure of ME/DE when compared to non-adaptive solutions, as depicted in Fig. 4c (content-adaptive versus *G0-only*, *G1-only* and *G2-only*) where the content-adaptive solution presents a more concentrated distribution near the zero value.

4. RESULTS AND DISCUSSIONS

The experimental results were generated on the JMVC 8.5 reference software [18] using the video sequences and common test conditions recommended by JVT [17]. Four video sequences with different spatial behavior and three different resolutions were used: *Ballroom* and *Flamenco2* (VGA-640x480), *Balloons* (XGA-1024-768) and *Poznan CarPark* (HD1080-1920x1088). The experiments were performed using 4-views sequences, IBP structure [3], CABAC, FRExt, *QP={22,27,32,37}*, GOP=8, TZ Search [2].

4.1 External Memory Bandwidth Savings

Fig. 5 presents the external memory bandwidth savings of the proposed DPB compression scheme, considering three components: (a) search window fetching (read), (b) reconstructed samples storage (write) and (c) error transmission (read/write) (to avoid encoder/decoder reference mismatches). The results consider two scenarios: Level C data reuse (a) without any compression and (b) with the proposed content-adaptive DPB compressor scheme for several video sequences.

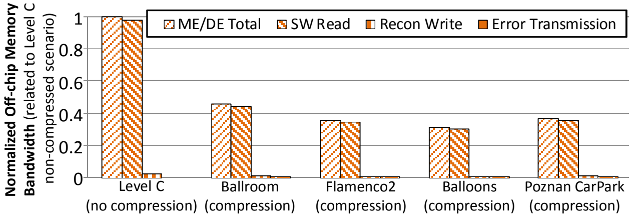


Fig. 5. Off-chip memory bandwidth savings

On average, the proposed content-adaptive reference compression is able to save 63% external memory access, when applied to the Level C data reuse [8]. Moreover, the block-based processing nature of our scheme has a perfect match to regular access data reuse schemes, like Level C. As depicted in Fig. 5, the ME/DE read operation to fetch the search window, even using the Level C data reuse strategy, is responsible for about 97% of the DPB external memory bandwidth. In turn, the error transmission overhead, for recovering the original reference at the decoder side, requires a negligible external memory bandwidth, about 1.5% of the total communication.

4.2 R-D Efficiency Gains Over Static Approaches

Fig. 6 presents the rate-distortion curves which compare the proposed content adaptation strategy against non-adaptive approaches: *G0-only*, *G1-only*, *G2-only* and *G3-only* (lossless). The non-adaptive approaches use the same residue compression parameters (quantization strength and Huffman table) in the entire encoding process.

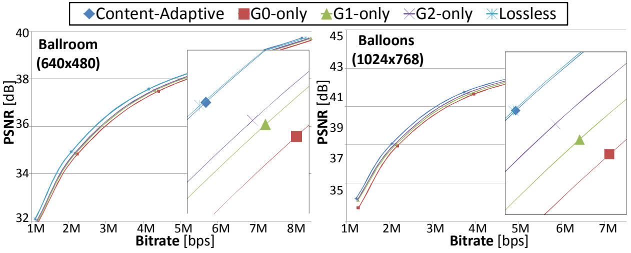


Fig. 6. Rate-distortion comparison (static vs. adaptive)

It can be noted that the proposed content-adaptive compression technique surpasses the rate-distortion efficiency of all lossy approaches in Fig. 6. As already discussed in Fig. 4c, the content-adaptation minimizes the error propagation along the prediction structure, since the future ME/DE will have better quality references. Regarding the lossless corner case, the proposed content-adaptive scheme causes negligible losses in rate-distortion efficiency. Using the Bjontegaard Delta metrics, BD-PSNR and BD-BR [20], the rate-distortion variations were -0.01dB and +0.18%, respectively.

4.3 Comparisons with State-of-the-Art

Tab. 1 presents the comparison of our content-adaptive scheme with state-of-the-art [12], [13], [14] and [15]. It can be noted that none of the works have performed evaluations considering the MVC encoding. In this sense, the error propagation path along the ME/DE references considered in the related works is shorter than that considered in this work, due to the multiple view coding and complex prediction structures.

Tab. 1. MVC Encoder Performance Evaluation (compared to related works)

Parameter	Lossy				Lossless	
	Our *(content-adaptive)*	**[12]**	**[13]**	**[14]**	**Our *(G3-only)***	**[15]**
Target	**MVC**	non-MVC	non-MVC	non-MVC	**MVC**	non-MVC
Content-Adaptive?	**Yes**	Yes	No	No	**No**	No
External Memory Savings	**69.5%**	25-50%	21-31%	17-24%	**51.3%**	24%
BD-PSNR	**-0.01 dB**	-0.04 dB	*N.I.*	-0.01 dB	**0 dB**	0 dB
BD-BR	**0.18%**	1.36-3.92%	0.38-21%	0.7%	**0%**	0%

Tab. 1 shows that our content-adaptive reference compressor is able to reduce the error propagation to achieve as negligible rate-distortion drops as the related single-view DPB compressor algorithms [13], [14] and [12]. Our content adaptation surpasses the adaptive scheme of [12] in all aspects: i.e., external memory reduction (39% of savings) and rate-distortion efficiency (0.03dB increased BD-PSNR). An additional column in Tab. 1 was inserted to compare [15] with a lossless non-adaptive version of our scheme using only the *G3* configuration. In this comparison, even without content adaptation, our scheme achieves the best results, surpassing the related work external memory savings of [15] by 27.3%.

5. CONCLUSIONS

This work presented a content-adaptive reference frame compressor to deal with external memory access issues in Multiview Video Coding due to significant memory requirements of reference frames of multiple views. Our scheme exploits the spatial redundancy by using a simplified intra-predictor. The prediction residue is then compressed by a content-adaptive compression path, composed of non-linear quantization and Huffman-based entropy encoder. In order to jointly minimize the external memory access and the MVC rate-distortion losses, a strategy to dynamically adapt quantization strength and Huffman tables according to the video content was proposed. Our scheme provides up to 63% external memory access savings along with negligible losses in rate-distortion efficiency. The external memory access reduction along with minimal drops in the MVC encoding efficiency demonstrated the high potential of our proposed scheme in next-generation of 3D-devices.

6. REFERENCES

[1] A. Smolic, et al. “Coding Algorithms for 3DTV - A Survey.” In: IEEE TCSVT, v. 17, n. 11, pp. 1606-1621, nov. 2007.

[2] Joint Draft 8.0 on Multiview Video Coding, JVT-AB204, 2008.

[3] P. Merkle, et al. “Efficient Prediction Structures for Multiview Video Coding.” In: IEEE TCSVT, v. 17, n. 11, pp. 1461-1473, nov. 2007.

[4] B. Zatt, M. Shafique, F. Sampaio, L. Agostini, S. Bampi, J. Henkel, "Run-time adaptive energy-aware motion and disparity estimation in multiview video coding", IEEE DAC, pp. 1026-1031, 2011.

[5] B. Zatt, M. Shafique, S. Bampi, J. Henkel, “A Low-Power Memory Architecture with Application-Aware Power Management for Motion & Disparity Estimation in Multiview Video Coding”, IEEE ICCAD, pp. 40-47, 2011.

[6] M. Shafique, B. Zatt, F. L. Walter, S. Bampi, J. Henkel, “Adaptive Power Management of On-Chip Video Mamory for Multiview Video Coding”, IEEE DAC, pp. 866-875, 2012.

[7] J.-C. Tuan, et al. “On the Data Reuse and Memory Bandwidth Analysis for Full-Search Block-Matching VLSI Architecture.” In: IEEE TCSVT, v. 12, n. 1, p. 61-72, jan. 2002.

[8] C.-Y. Chen, et al. “Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders.” In: TCSVT, v. 16, n. 4, p. 553-558, april. 2006.

[9] P.-K. Tsung, et al. “System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint”. IEEE ISCAS p. 1001-1004, 2007.

[10] T.-C. Chen, et al, “Single Reference Frame Multiple Current Macroblocks Scheme for Multi-Frame Motion Estimation in H.264/AVC”, In IEEE ISCAS, pp. 1790 – 1793, 2005.

[11] Agilent, “DDR Memory Design and Test Overview”. Available at: <http://www.home.agilent.com/>, 2012.

[12] L. Song, et al. "An adaptive bandwidth reduction scheme for video coding", In: IEEE ISCAS, pp.401-404, 2010.

[13] Z. Ma and A. Segall. "Frame buffer compression for low-power video coding", In: IEEE ICIP, pp.757-760, 2011.

[14] A. Gupte, et al. "Memory Bandwidth and Power Reduction Using Lossy Reference Frame Compression in Video Encoding", In: IEEE TCSVT, v. 21, n.2, pp.225-230, Feb. 2011.

[15] D. Silveira, et al, "Memory bandwidth reduction in video coding systems through context adaptive lossless reference frame compression," In: SPL, pp.1-6, 2012.

[16] Z. Wang; et al, "Memory efficient lossless compression of image sequences with JPEG-LS and temporal prediction," In: PCS, pp.305-308, 2012.

[17] Huffman, D.A., "A Method for the Construction of Minimum-Redundancy Codes," IRE, vol.40, no.9, pp.1098-1101, Sept. 1952.

[18] JMVC Reference Software, Sep. 2009.

[19] JVT. “Com. Test Cond. for Multiview Video Coding”. JVT-T207, 2007.

[20] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, VCEG Contribution VCEG-M33, Austin, April 2001.

Run-Time Adaptive Energy-Aware Motion and Disparity Estimation in Multiview Video Coding

Bruno Zatt[1,2], Muhammad Shafique[1], Felipe Sampaio[2,3], Luciano Agostini[3], Sergio Bampi[2], Jörg Henkel[1]

[1]Karlsruhe Institute of Technology (KIT), Chair for Embedded Systems, Karlsruhe, Germany
[2]Federal University of Rio Grande do Sul (UFRGS), Informatics Institute/PGMICRO, Porto Alegre, Brazil
[3]Federal University of Pelotas (UFPel), GACI, Pelotas, Brazil

{bzatt, bampi, felipe.sampaio}@inf.ufrgs.br, agostini@inf.ufpel.edu.br, {muhammad.shafique, henkel}@kit.edu

ABSTRACT

This paper presents a novel run-time adaptive energy-aware Motion and Disparity Estimation (ME, DE) architecture for Multiview Video Coding (MVC). It incorporates efficient memory access and data prefetching techniques for jointly reducing the on/off-chip memory energy consumption. A dynamically expanding search window is constructed at run time to reduce the off-chip memory accesses. Considering the multi-stage processing nature of advanced fast ME/DE schemes, a reduced-sized multi-bank on-chip memory is employed which can be power-gated depending upon the video properties. As a result, when tested for various video sequence, our approach provides a dynamic energy reduction of 82-96% for the off-chip memory and a leakage energy reduction of 57-75% for the on-chip memory compared to the Level-C and Level-C+ [7] prefetching techniques (which are the prominent data reuse and prefetching techniques in ME for video coding). The proposed ME/DE architecture is synthesized using a 65nm IBM low power technology. Compared to state-of-the-art MVC ME/DE hardware [14], our architecture provides 66% and 72% reduction in the area and power consumption, respectively. Moreover, our scheme achieves 30fps ME/DE 4-view HD1080p encoding with a power consumption of 74mW.

Categories and Subject Descriptors: C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; B.3.2 [**Design Styles**]: Cache memories; I.4.2 [**Compression (Coding)**]: Approximate methods

General Terms: Algorithms, Design, Management

Keywords: MVC, Video Coding, Motion and Disparity Estimation, Energy-Aware Design, On-Chip Memory

1. INTRODUCTION AND MOTIVATION

The increasing consumer interest in new immersive/3D multimedia technologies (based on multiple views) have led to the evolution of 3D personal video recording and playback for the next-generation mobile devices [1]. Early case studies on 3D-camcorders and 3D-mobile phones have demonstrated the feasibility of multiview video recording on mobile devices [2][3]. Due to the battery-powered nature of such mobile devices, energy reduction is one of the primary design goals. However, it is a grand research challenge in the existence of enormous data fetching and processing due to multiview videos.

The Multiview Video Coding (MVC) [4] has emerged as a new coding standard for 3D-videos. It provides double compression compared to the H.264 video encoder at same quality [1] by exploiting the inter-view redundancy. This improved compression comes at the cost of significantly increased energy consumption due to variable-sized Motion and Disparity Estimation (ME, DE) that exploit temporal and inter-view redundancies, respectively. This poses a serious challenge on the realization of real-time high-definition MVC on battery-powered devices.

Typically a Macroblock (MB) is searched in a pre-defined *Search Window* in a reference frame/view through a block matching process using SAD (Sum of Absolute Differences). Fig. 1 demonstrates our energy analysis of MVC for the "Ballroom" sequence (640x480, with high motion) for various search window sizes[1]. It can be seen that the ME and DE may consume up to 98% of the total encoding energy. Besides the SAD computations, the increased energy is mainly due to the enormous memory accesses to perform the memory-intensive block-matching process in ME and DE. Fig. 1 illustrates that the energy consumption by the on-chip storage and off-chip memory accesses may exceed up to 90% of the total ME & DE energy. Such an analysis for ME in MPEG standards have been demonstrated in [17], which also states a similar finding. However, in case of MVC, DE results in a significant increase in the memory energy consumption, because, typically larger search windows are deployed for DE. Therefore, *there is a dire need to reduce the energy of the on-chip memory (that stores the search window) and off-chip memory accesses in ME and DE* in order to realize real-time high-definition MVC on battery-powered devices.

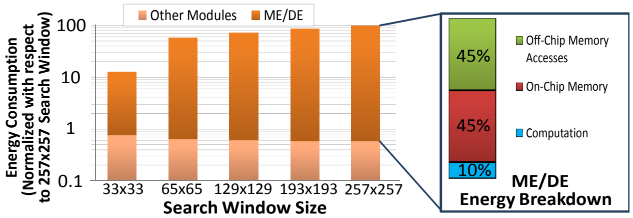


Fig. 1 Energy consumption breakdown for computation, on-chip storage and off-chip communication

To reduce the memory energy, state-of-the-art techniques [7][9][12]-[15] typically employ search window prefetching for avoiding frequent off-chip memory accesses (as the pixels in the search window are accessed multiple times). These techniques use an on-chip memory to store the pre-defined *rectangular* search windows. However, our ME/DE memory analysis (see details in Section 3.1) has demonstrated that several (pixel) regions in the rectangular search windows are not used for SAD computations. It is mainly due to the adaptive nature of advanced ME/DE schemes [18][2]. The advanced fast ME/DE schemes find the best match (i.e., block with the minimum SAD in the search window) using multiple search stages, where the candidate search point with minimum SAD in a preceding search stage serves as starting point for the succeeding search stage. In this way, fast ME/DE schemes

DAC'11, June 5-10, 2011, San Diego, California, USA

[1] Note, a fast ME/DE algorithm (*TZ Search* [18]) is considered in these experiments, which is up to 23x faster than the *Full Search*.

[2] Such advanced adaptive ME schemes are required to reduce the number of search candidates (i.e., SAD computations and memory accesses).

move in a certain direction to find the global minimum. Moreover, fast ME/DE schemes significantly reduce the number of search candidates that may vary significantly depending upon the properties of an MB. As a result, several regions of the search window (that are not on the search trajectory of the ME scheme) may never be used for SAD computations. Significant energy savings for the off-chip memory accesses can be obtained by avoiding the prefetching of unused regions of the search window. Therefore, *an adaptive search window formation is required such that the search window is formed at run time depending upon the behavior of the ME scheme and the trajectory*.

For deep sub-micron fabrication technologies, leakage consideration becomes imperative in the energy-efficient design of ME/DE. One key source of leakage is the big on-chip SRAM memory to store the search window of bigger size, which is inevitable in case of DE. The unused regions of the rectangular search window indicate wastage of on-chip memory hardware. Therefore, *significant leakage reduction may be obtained by reducing the size of the on-chip memory, while considering an analysis of the memory requirements of fast ME/DE schemes*.

Even for a given reduced-sized on-chip memory, the complete memory may not be used by all of the MBs. It is due to the fact that different MBs exhibit diverse texture and motion properties. The fast ME/DE schemes adapt their search patterns and employ early termination depending upon the properties of an MB, thus requiring different amount of data (from the search windows). Therefore, there is a need for a multi-bank on-chip memory organization where different banks may be power-gated. Typically, state-of-the-art employs power-gating depending upon the idle state period of the hardware. However, the amount of data required for the ME/DE may already be predicted by considering the properties of an MB and the ME/DE scheme. Therefore, *there is a need to raise the abstraction level of the power shutdown decision to the application level*. This provides a much higher potential for leakage energy savings.

Summarizing the above discussion, **the key research challenges** in the energy-aware design of ME/DE in MVC are:

a) Adaptively forming the search window at run time depending upon the trajectory of fast ME/DE scheme and MB properties
b) Determining an appropriate size and multi-bank organization of the on-chip memory for adaptive search window
c) Raising the abstraction level of the power shutdown decision to the application level and enabling an application-aware power-gating control of different banks of the on-chip memory

1.1 Our Basic Idea and Novel Contribution

To address the pointed challenges, *a novel run-time adaptive energy-aware Motion and Disparity Estimation architecture* is proposed.

1) ***A Dynamically Expanding Search Window Formation Algorithm:*** Instead of prefetching the complete rectangular search window, a selected partial window is formed and prefetched for each search stage of a given fast ME/DE scheme depending upon the search trajectory, i.e., the search window is dynamically expanded depending upon the outcome of each search stage. An analysis is performed to highlight the issues related to the expansion of the partial window at each search stage. The search trajectories of the neighboring MBs and their spatial and temporal properties (variance, SAD, motion and disparity vectors) [21] are considered to predict (at run time) the form of the search window for the current MB. This results in a significantly reduced energy for off-chip memory accesses.
2) ***A Hardware architecture with Multi-Bank On-Chip Memory:*** A hardware architecture with parallel SAD modules is proposed. A pipelined schedule is proposed to enable high throughput. Moreover, the hardware is equipped with a multi-bank on-chip memory to provide high throughput in order to meet HD requirements. The size and the organization of the memory is obtained by an analysis of the fast ME/DE scheme. Each bank is partitioned into multiple sectors, such that each sector can be individually power-gated to save leakage. The control of the power-gates is obtained from the application layer.
3) ***An Application-Aware Power-Gating Scheme for the On-Chip Memory:*** Depending upon the fast ME/DE scheme and the Macroblock properties, the amount of required data is predicted. Only the sectors to store the required data are kept powered-on and the remaining sectors are power-gated.

To the best of our knowledge, this is the first joint ME/DE architecture for MVC that employs dynamically expanding search windows and application-aware power-gating control to reduce the memory energy, while considering the adaptive nature of advanced fast ME/DE schemes and diverse video properties. The proposed architecture requires the knowledge of a given fast ME/DE scheme at design time to perform a memory access analysis, though the concept is not fixed to any specific ME/DE scheme.

The Paper Organization: Section 2 discusses the related work. Section 3 presents the memory access pattern analysis and the algorithm for dynamically expanding search windows. Section 4 presents the ME/DE architecture with the design of multi-bank on-chip memory and application-aware power-gating control. Section 5 discusses the results followed by the conclusion in Section 6.

2. RELATED WORK

In [5] a view-adaptive algorithm for low complexity MVC encoding is proposed. In [20] is presented an energy budgeting scheme for the H.264 ME. The authors in [7] evaluated different data reuse schemes and proposed a new search window-level reuse scheme for H.264 ME. In [8] a bandwidth-efficient H.264 ME architecture using binary search is proposed. The MVC encoder presented in [9] employs a search window prefetching technique to feed the ME/DE hardware unit. However, these search window-based approaches suffer from the excessive leakage of their big on-chip SRAM memories in order to store the complete rectangular search windows. This point becomes crucial for MVC as the DE requires relatively large search windows (mainly for high resolutions) such as [±96,±96] to accurately predict high disparity regions [10]. In this case, even considering asymmetric search windows incur large on-chip storage overhead, thus suffering from significant leakage (considering deep sub-micron technologies). The authors in [11] use multiple on-chip memories to realize a big logical memory or multiple memories (one for each reference frame) according to the frame motion. A data-adaptive structured search window scheme is presented in [12]. It is based on window-follower approach for H.264. The work in [13] proposed a candidate-level data reuse scheme and a Four Stage Search algorithm for ME. In [14] a cache-based ME/DE is presented proposing a search range reduction by predicting the center of search window based on the neighborhood. However, the authors ignored the fact that fast ME/DE schemes already consider this information to start the search. The work in [22] proposes a fast ME/DE scheme for MVC and a multi-level pipelined architecture with multiple caches. Finally, in [15] a caching algorithm is proposed for fast ME. Additionally, a prefetching algorithm based on search path prediction is proposed in order to reduce the number of cache misses. The work [15], however, it is limited to a fixed Four Step Search pattern and it does not consider disparity estimation and power-gating.

Our approach is different from state-of-the-art as it dynamically expands the search window at run time based on the search trajectories of the neighboring MBs. It employs a much smaller on-chip memory based on a compile-time analysis and it additionally power-gates the unused sectors of the on-chip memory considering the MBs properties.

3. DYNAMICALLY EXPANDING MOTION & DISPARITY SEARCH WINDOW FORMATION

3.1 ME/DE Memory Access Analysis

The design of the dynamically expanding search windows is based on the following analysis of the memory access patterns of different fast ME/DE schemes in temporal and view domains. Real-world embedded systems typically deploy a fast ME/DE scheme (like *TZ Search* [18] [19] and *Log Search* [19], instead of a *Full Search* which is impracticable due to its large computation and energy requirements). These schemes are based on multiple search stages with a set of initial search point predictors, multiple search patterns, and early termination criteria. The key idea of such schemes is to iteratively move towards the global minimum by following the trajectory of search candidates with minimum SAD at each search stage. As a result, these schemes evaluate a significantly reduced number of search candidates, thus requiring a considerably reduced portion of the search window (mainly in the direction of the search trajectory).

Fig. 2 shows the ME process for two Macroblocks (MBs, in the Ballroom video sequence) with low and high motion using the *TZ* and *Log Searches* [18][19] along with the accessed pixels (in purple) within the search window (shown as a white rectangular box). In case of a high motion MB (dancers), *TZ Search* processes multiple search pattern stages in order to find a better match, as it can be seen by the displaced diamond patterns forming a search trajectory. In contrast to this, for a low motion MB (background and spectators), TZ Search mainly converges inwards (i.e., towards the centre of the search window) by switching from large diamond pattern to smaller ones, thus using significantly less area of the search window. The analysis in Fig. 2 shows that the amount of unused pixels in the slow motion MB is much less than that in the high motion MB. Even for the high motion MB large portions of the search window are unused. A similar behavior was observed in the DE process, too. Fig. 3 shows the search window wastage (i.e., the amount of data fetched from the off-chip memory to the on-chip memory in a search window-based prefetching that is not used by the search algorithm) for various video sequences with diverse motion and disparity characteristics. Fig. 3 shows that the search window wastage may exceed 80% of the total search window area that directly corresponds to high energy wastage for the off-chip memory accesses and inefficient utilization of the on-chip memory resources. In case of DE, the wastage is relatively less, as the *TZ* scheme performs a bigger search to capture the longer disparity vectors. Fig. 2 and Fig. 3 show that the amount of unused pixels in the search window is much more in case of the *Log Search* (up to 95%). It is due to the search algorithm and pattern structure of the *Log Search*. However, *TZ Search* provides a better video quality compared to the *Log Search*.

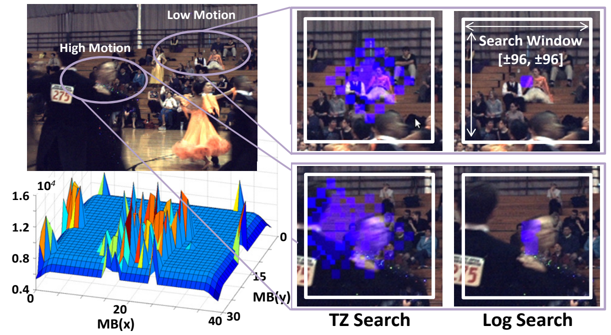


Fig. 2 ME/DE Search pattern for TZ Search and Log Search; Plot shows the Number of Pixels Accessed in the External Memory

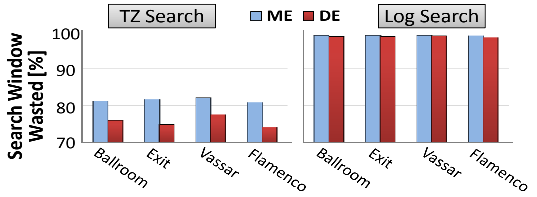


Fig. 3 ME/DE Search Window Wastage for Various Sequences

This analysis demonstrates significant energy reductions may be performed by avoiding the prefetching of the complete search window (that may incur a significant initial stall before processing SADs). Rather, a much smaller portion of the search window needs to be prefetched while considering the search pattern shapes and different search stages of a fast ME/DE scheme. Moreover, this analysis demonstrates the significance to consider MB properties (background or moving objects) and search pattern type in predicting the shape (also the expansion direction) and the size of useful search window regions. Furthermore, the on-chip memory can be partitioned into sectors, such that each sector can be individually clock- and/or power-gated depending upon the (search window) data requirements of an MB (that depends upon its spatial, temporal, and disparity properties, such as texture, SAD, motion & disparity field). Fig. 2 shows that the peak search window usage is in the MBs of the dancers and around the corners (where new objects are appearing or disappearing). Remaining slow motion or stationary MBs (of the spectators and background/floor) require a significantly reduced portion of the search window. It is worthy to note in the surface plot that the MBs of the same object require almost the same data from the search window for ME/DE, i.e., they tend to have similar memory access behavior. This is mainly due to the fact that MBs of the same object tend to move in the same direction, thus exhibit similar motion and disparity properties. Therefore, the so-called *Search Map* (i.e., the dynamically formed search window shape and size) of the neighboring MBs may be used to predict the *Search Map* of the current MB, considering they share the similar search trajectory. Moreover, MBs of the background and stationary regions typically demonstrate inward search trajectory, thus an inwards, converging *Search Map*. Note, in addition to the spatial properties of an MB (texture measured with variance), the temporal and disparity fields help to predict the *Search Map*. A change happens at the object boundaries that require a special care to be taken. In the following, we will discuss how a *Search Map* may be predicted.

3.2 Search Map Prediction

Fig. 4 presents the *Search Map* for two neighboring MBs (denoted as MB_x and MB_{x+1}) using the *Log Search* algorithm. After the ME search is performed for the MB_x, a *Search Map* is built based on the search trajectory (i.e., the ID of the selected candidate search points at each search stage of the ME/DE scheme). As shown in Fig. 4a, the first search stage selects the candidate with ID 6 as the best candidate. Similarly, candidates with ID 3 (at stage 2)[3], ID 4 (at stage 3), and ID 4 (at stage 4) are selected as the best candidates at their respective search stages. This provides a *Search Map* of [6,3,4,4] (the trajectory is shown by the red arrows). Note, for each search stage there is an entry in the *Search Map* with the ID of the candidate with minimum SAD at that particular search stage.

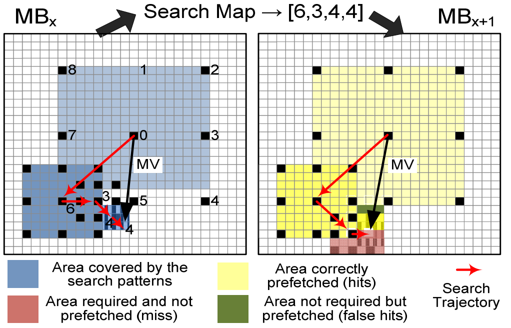

MB_x
Search Map → [6,3,4,4]
MB_{x+1}
8
1
2
7
0
3
MV
6
3
5
4
4
MV
Area covered by the search patterns
Area correctly prefetched (hits)
Search Trajectory
Area required and not prefetched (miss)
Area not required but prefetched (false hits)

Fig. 4 Search Map Prediction for the Log Search

Considering the above analysis of the MB neighborhood (see Section 3.1), a *Search Map* for the MB_{x+1} can be predicted from the *Search Map* of MB_x. In case there is a deviation in the search trajectory of these two MBs, there will be a miss in the on-chip memory due to

[3] In each next search stage, the IDs of the search candidates are reset; the index of the ID in the *Search Map* denotes the search stage number.

the prefetch of the false region (see the Green box in Fig. 4b). Typically, these misses are at the boundaries of the moving objects and are relatively in very few MBs in the whole video frame. In case of a miss, there will be a stall for only the prefetching of the first candidate's data on the new trajectory (i.e., 16x16 pixel data). All other candidates on the new trajectory will be then prefetched correctly (before their respective SAD computations, thus not causing any stall) as the search pattern design of the fast ME/DE schemes is fixed at design time (see the brown box for the new prefetched data). Typically a miss in the trajectory depends upon the motion/disparity difference of two MBs, which is significantly smaller in most of the neighboring MBs due to high correlation between them (see the plot in Fig. 2, Section 3.1).

3.3 Algorithm for the Dynamically Expanding Search Window Formation

Fig. 5 shows the pseudo-code for the algorithm of the dynamic search window formation and expansion. The algorithm works in two major steps. First it predicts the *Search Map* from the spatial predictors (lines 3-21). Afterwards, it checks if the search pattern matches the *Search Map* or not and it prefetches the appropriate partial search window and updates the *Search Map* (lines 23-33).

```
1.  // Predict the Search Map from the Neighboring MBs
2.  PredictorSet ← Ø;
3.  If (PredictorsAvailable) Then
4.     PredictorSet = {$MV_{Left}$, $MV_{Top}$, $MV_{TopRight}$, $MV_{SpatialMedian}$};
5.     computeVariance (PredictorSet); //Compute Variance of all predictors
6.     getTemporalInfo (PredictorSet, currMB); //Get MV, DV, SADs
7.     $Motion_{currMB} = (SAD_{currMB} > TH_{SAD})?\ 1: 0;$
8.     For i = 0 to all Predictors //Compute the Similarity of predictors,i.e.,
       check if predictors belong to the same object as of the current MB
9.        $diffVar_{predi} = Var_{currMB} - Var_{predi};$
10.       $Motion_{predi} = (SAD_{predi} > TH_{SAD})?\ 1: 0;$
11.       $diffMotion_{predi} = Motion_{currMB} - Motion_{predi};$
12.       $predDiff_{predi} = \alpha * diffVar_{predi} + \beta * diffMotion_{predi};$
13.    End For
14.    PredictorSet = sortPredictors (predDiff, PredictorSet);
15.    bestPred = determineBestPredictor (PredictorSet, currMB);
16.    If ($predDiff_{bestPred} < TH_{diff}$) Then
17.       $predSearchMap = SearchMap_{bestPred};$
18.    Else
19.       predSearchMap = findClosestSM (PredictorSet, $TH_{diff}$);
20.    End If
21. End If
22. // Perform Dynamic Formation and Expansion of the Search Window
23. For all SearchStages          // Depending upon the  fast ME/DE scheme
24. SM_Miss = checkSearchMap (searchStageID, predSearchMap);
25.    If ((PredictorSet == Ø) or SM_Miss) Then
26.       SWBuffer = prefetchPartialWindow (refFrame, searchStageID,
                                            searchStagePattern);
27.    Else
28.       SWBuffer = prefetchPartialWindow (refFrame, searchStageID,
                                            predSearchMap);
29.    End If
30.    bestCand = performMEDE (currMB, SWBuffer, SearchAlgorithm);
31.    Build_CurrMB_SearchMap (bestCand, searchStageID);
32.    If (earlyTermination)      return;      End If
33. End For
34. return;
```

Fig. 5 Algorithm for Search Map Prediction and the Dynamic Formation and Expansion of the Search Window

For predicting the *Search Map* four spatial predictors are considered that have high correlation with the current MB (line 4). Afterwards, variance of these predictors is computed and motion and disparity information is obtained (lines 5-6). Based on the spatial, temporal, and view information, a matching function is computed that provides a hint that predictors may belong to the same object or may exhibit similar motion/disparity properties (lines 7, 9-12). Afterwards, the predictors are sorted w.r.t. their similarity to the current MB (line 14). The closest predictor is determined by computing the SAD with the current MB (line 15). In case the closest predictor also belongs to the same object or exhibit similar motion/disparity, its *Search Map* is considered as the predicted *Search Map* (line 17). Otherwise, the closest map is found in the remaining predictor set (line 19). If none of the predictors exhibit similarity to the current MB, then the predicted *Search Map* is empty.

After the *Search Map* is predicted, it is used to construct search window. For each search stage, the partial search window is determined according the *Search Map* and prefetched. In case the search candidates of the search pattern are present in the *Search Map* (i.e., the search trajectory falls in the predicted region), the partial search window is simply constructed according to the predicted *Search Map* and the prefetched data is used (i.e., a case of *hit*); see line 28. Otherwise, if the *Search Map* is empty or does not contain the search candidate, the *Search Map* is ignored (for this stage and onwards); see line 26. In this case the prefetched data is wasted and it is considered as a *miss*. The partial search window is then constructed according to the search pattern for the miss parts (see line 31, it can also be seen in the example of Fig. 4b).

Now we will discuss the architecture of our joint ME/DE scheme along with the design of the multi-bank on-chip memory (to store the search window) and application-aware power-gating.

4. ARCHITECTURE OF ENERGY-AWARE MOTION AND DISPARITY ESTIMATION

Fig. 6 shows the hardware architecture of our proposed run-time adaptive energy-aware Motion and Disparity Estimation (ME, DE) scheme. It employs the above-discussed dynamically expanding search window prefetching and a multi-banked on-chip memory with application-aware power gating control. In order to obtain high throughput, a set of 64 (4-pixel) SAD operators and 21 SAD trees is provided as computation block. Each SAD operator requires 4 pixels from current frame and 4 pixels from reference (stored in the on-chip search window memory). A ME/DE search control unit is integrated which can be programmed to realize various fast ME/DE schemes. This unit controls the search stages and patterns, and it provides the required algorithmic information to various other modules. The search window formation unit predicts the Search Map and dynamically constructs the search window structure. This data corresponding to the window is prefetched in the multi-bank search window memory which consists of various sectors that can be individually power-gated depending upon the ME/DE requirements of the current MB.

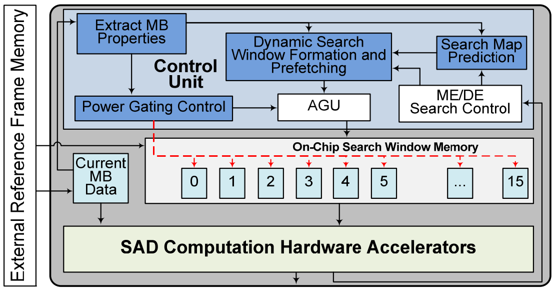

External Reference Frame Memory
Extract MB Properties
Dynamic Search Window Formation and Prefetching
Search Map Prediction
Control Unit
Power Gating Control
AGU
ME/DE Search Control
On-Chip Search Window Memory
Current MB Data
0
1
2
3
4
5
...
15
SAD Computation Hardware Accelerators

Fig. 6 ME/DE Hardware Architecture Block Diagram

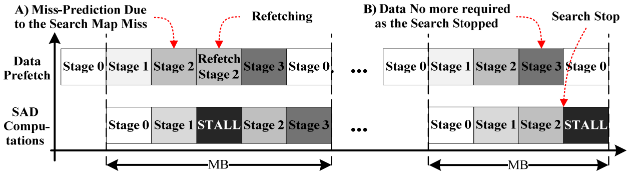


Fig. 7 Pipeline Processing Schedule of our ME/DE Architecture

Fig. 7 presents the MB-level ME/DE processing pipeline showing the data prefetching and SAD computation for different search stages. During the SAD computations of the preceding search stage, the partial search window data is prefetched for the succeeding search stage. However, in case of a Search Map miss, stall for one candidate data prefetch happens (see *A* in Fig. 7). In case the fast ME/DE scheme stops the search due to early termination criteria, the prefetch data in the search window is wasted (see *B* in Fig. 7). Now we discuss the design of the design and power-gating control for the on-chip memory to store the search window.

4.1 Memory Hierarchy

The size of the dynamically formed search window is significantly less compared to the rectangular search windows (like in [7][9][12]-[15]). Rectangular windows incur increased area and leakage of on-chip memory. This scenario becomes even more critical in MVC where ME and DE are performed for multiple views using larger search windows (for instance 193x193 to capture high disparity regions in DE). Depending upon the MB properties, the sizes of dynamically expanding search windows may vary significantly. However, the size of on-chip memory that stores this window must be fixed at design time. Therefore, we first perform an exploration to obtain a reasonable size of the on-chip memory (that provides leakage reduction and area savings). In case the MB exhibit low motion and the size of the prefetched window is still less than the on-chip memory, the remaining parts of the memory are power-gated to further reduce leakage.

Fig. 8 demonstrates our exploration of the memory access distribution for the *Ballroom* (a fast motion sequence). Fig. 8a shows the number of MBs for which ME and DE process less than 96 MBs. Please note that the reduced number is mainly due to the adaptive nature of fast ME/DE schemes and it does not mean that this is within a smaller search range. A rectangular search window of 193x193 size requires 37KB of on-chip memory. Fig. 8b shows that even for such a large search range (i.e., a search window of 193x193), at most 96 candidates are evaluated per MB. This corresponds to an on-chip memory of 24KB, i.e., a reduction of 35% area. When scrutinizing the Fig. 8b, it is noticed that in more than 95% cases a storage of only 64MBs is required (i.e., 16 KB → 57% savings). We have performed such an analysis for various video sequences with diverse motion (not shown here due to space reasons). Similar observations were made in all of the cases. Therefore, we have selected an on-chip memory of 16KB, which provides significant leakage reduction in the on-chip memory. In few rare cases, where the ME and DE may require more MBs, misses may happen (as we will show in the result section). To provide high parallelism, the on-chip memory is organized in 16 banks, where one 16 pixel row of an MB is stored in each of the banks.

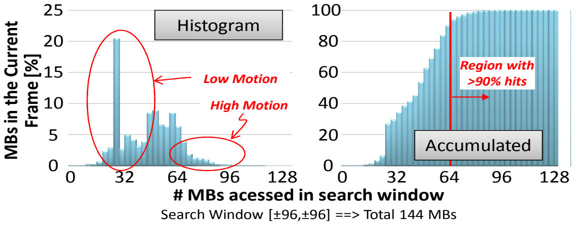


Fig. 8 Analyzing the Memory Requirements for ME/DE of Different MBs in the Ballroom Sequence

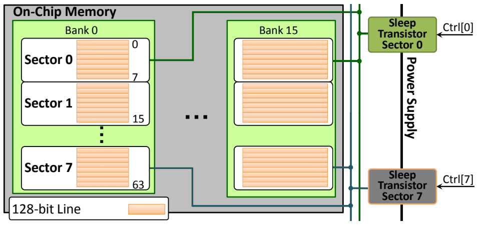


Fig. 9 Search Window Memory Organization with Power-Gating

As discussed above, even 16KB memory may not be completely used to store the dynamically expanding search window as the size of prefetched search window highly depends upon the MB properties and fast ME/DE scheme (as seen in Fig. 8b that in more than 20% of the cases storage for 32MBs is used, i.e., half of the memory). Therefore, each bank is partitioned into multiple sectors[4] where each sector can be individually power-gated to further reduce the leakage (see Fig. 9). The main challenge here is to incorporate the application and video knowledge to determine the power-gate control, such that signals to the power-gates may be determined depending upon the predicted memory requirements of the current MB.

4.2 Application-Aware Power-Gating Control

Considering the MB properties and ME/DE scheme provides a relatively high potential for leakage reduction by predicting the memory requirements of MBs before their actual ME/DE. However, frequent on-off switching needs to be avoided as power-gating incurs wakeup energy overhead. Therefore, our scheme predicts the sleep time as function of *'n'* consecutive MBs for which sectors of the on-chip memory can be power-gated. Considering the worst case of stationary MBs, to overcome the wakeup energy overhead, the following condition must hold: $P_{leak\_onChipMem} * T_{minMEDE} * n > E_{wakeup}$.

However, the minimum ME/DE time depends upon the deployed fast ME/DE scheme. For instance, in case of a stationary MB there will be a minimum of 9 SAD computations for the *Log Search* and for the *TZ Search* it is 46 SADs. Therefore, considering the minimum number of SADs for a stationary MB, the above equation can be re-written as: $n > (E_{wakeup} / P_{leak\_onChipMem} * minNumberSADs * T_{SAD})$; where *minNumberSADs* is 9 and 46 for *Log* and *TZ* Searches, respectively. For a given sleep transistor design and a given SRAM memory, the *n* can be determined. In reality, MBs exhibit diverse motion and spatial properties. Therefore, the number of consecutive MBs that require a certain amount of on-chip memory may even be less than *n*.

Let's assume *n* consecutive MBs require at most *R* KB of on-chip memory for their search window prefetching. For a given on-chip memory of size S_{memory} KB with N_{Sec} number of S_{Sec} KB sectors, the amount of power-gateable memory is computed as:

$$N_{gateableSectors} = (S_{memory} - R) / S_{Sec}$$

5. RESULTS AND EVALUATION

For the experimental results, a set of four video sequences each with 4 views is used. The search algorithms used are *TZ Search* [18] and *Log Search* [19] considering three QP values (22, 27, 32) and search in the four possible directions with a search window of 193x193. The thresholds set used was: N=6, α=1, β=500 and TH_{SAD}=400.

5.1 Synthesis Results

TABLE I presents the comparison of the ASIC implementation of our architecture with one of the most prominent related work [14]. Our design reduces the area and power consumption by 66% and 72%, respectively, while providing higher throughput compared to [14]. The hardware implementation executes at 300MHz and provides real-time ME/DE for up to 4-views HD1080p. This significant power reduction is mainly due to the employment of dynamic search window formation, power-gating, smaller logic, and fast ME/DE scheme. The authors of [14] use a data reuse like Level-C [7], which compared to our proposed solution performs inefficient as we will discuss. The on-chip memory in our hardware is relatively larger as it supports a much bigger search window of up to 193x193 compared to 33x33 in [14] (which is insufficient to capture larger disparity vectors). Note, Level-C [7] with a search window of 193x193 would require four memories of 288Kb (i.e., a total of 1,115 Mb) to exploit the reusability in four possible prediction directions available in MVC as our approach does with 512Kbits. To perform a fair comparison, we have deployed the Level-C and Level-C+ [7] techniques in our hardware architecture. Fig. 10 shows the energy benefit of employing our dynamically expanding search window and multi-bank on-chip memory with power-gating.

[4] 8 sectors in our case as typical search patterns have candidates in multiple of 8.

Compared to Level-C and Level-C+ [7] prefetching techniques (based on rectangular search windows), our approach presents energy reduction in on-chip and off-chip memories as shown in Fig. 10. For a search window of 193x193, our approach provides an energy reduction of up to 82-96% and 57-75% for off-chip and on-chip memory access, respectively. These significant energy savings are due to the fact that Level-C and Level-C+ [7] suffer from a high data retransmission for every first MB in the row (even rows in Level-C+). Additionally, our approach provides higher data reuse and incurs reduced leakage due to a smaller on-chip memory and power-gating of the unused sectors.

TABLE I. COMPARISON OF OUR FAST ME/DE ALGORITHM

	Tsung'09 [14]	Fast ME/DE Architecture
Technology	TSMC 90nm Low Power LowK Cu	ST 65nm Low-Power 7 metal layer
Gate Count	230k	102k
SRAM	64 Kbits	512 Kbits
Max. Frequency	300 MHz	300 MHz
Power	265mW, 1.2v	74mW, 1.0v
Proc. Capability	4-views 720p	4-views HD1080p

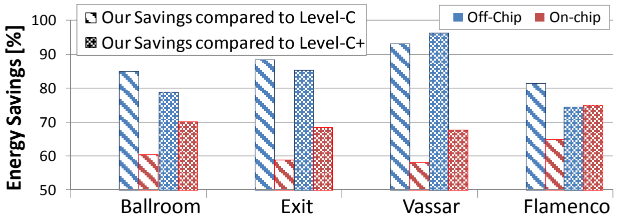


Fig. 10 On-chip and Off-chip Memory Energy Savings Compared to the Level-C and Level-C+ [7] Prefetching Techniques

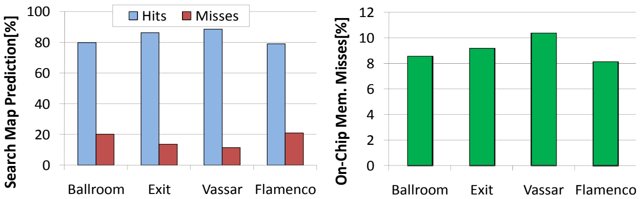


Fig. 11 Search Map Prediction & On-chip Memory Accuracy

Fig. 11 presents the evaluation of our scheme for *Search Map* misses and on-chip memory misses. Fig. 11 shows that the accuracy of Search Map prediction is higher for low motion sequences (e.g., Vassar) compared to high motion sequences (e.g., Flamenco) as the search trajectory is shorter and easier to be predicted (due to a higher number of stationary/slow-moving MBs). However, even for the worst case the hits are around 80% (see Fig. 11a). In case of off-chip memory accesses, the misses are higher for low motion sequences because the search trajectory tends to converge to the center (only the central region of search window is accessed) reducing the overlapping accessed area with the neighboring MBs. The higher number of memory misses for low motion sequences, however, does not contradict the higher off-chip energy savings achieved for the same sequences (as shown in Fig. 10). The reason is that the percentage of misses is calculated over a much smaller number of total memory accesses for low motion sequences. The measured miss-rate does not represent significant performance degradation in relation to [14] since the fast search deployed in our solution evaluates a reduced number of candidates. The quality loss of the *TZ Search* is insignificant compared to the exhaustive search [18].

6. CONCLUSIONS

We proposed a novel run-time adaptive energy-aware Motion and Disparity Estimation (ME, DE) architecture for MVC encoding. It integrates an efficient data prefetch technique, where the search window is constructed at run time depending upon the search trajectories of the neighboring MBs. The size of the on-chip memory is reduced by analyzing the storage requirements of the fast ME/DE schemes. Significant leakage energy savings are obtained by exploiting the application and video data information to perform power-gating of the unused parts of the on-chip memory. When compared to the prominent data reuse and prefetching techniques of ME in video coding (Level-C and Level-C+ [7]), the proposed architecture provides 82-96% and 57-75% energy reductions for off-chip and on-chip memories, respectively. The hardware architecture is implemented as an ASIC (using IBM 65nm low power technology) and it delivers a real-time ME/DE of up to four HD1080p views. The significant energy reduction demonstrates the higher potential of our proposed approach in next-generation mobile devices with 3D-video services.

7. ACKNOWLEDGMENT

We would like to thank Leandro Max de Lima, Federal University of Rio Grande do Sul, for his contribution to the hardware implementation.

8. REFERENCES

[1] P. Merkle et al., " Efficient Prediction Structures for Multiview Video Coding" IEEE TCSVT, vol.17, no.11, pp. 1461- 1473, 2007.

[2] FinePix REAL 3D W3 | FujiFilm Global: http://www.fujifilm.com/products/3d/camera/finepix_real3dw3/.

[3] Lynx 3D SH-03C: http://www.sharp.co.jp/products/sh03c/index.html

[4] Joint Draft 8.0 on Multiview video coding, JVT-AB204, 2008.

[5] L. Shen et al., "View-Adaptive Motion Estimation and Disparity Estimation for Low Complexity Multiview Video Coding", IEEE TCSVT, vol.20, no.6, pp.925-930, 2010.

[6] H.-C. Chang, et al., "A Dynamic Quality-Adjustable H.264 Video Encoder for Power-Aware Video Applications", IEEE TCSVT, vol.19, no.12, pp.1739-1754, Dec. 2009.

[7] C.-Y. Chen et al., "Level C+ data reuse scheme for motion estimation with corresponding coding orders", IEEE TCSVT, vol.16, no.4, 2006.

[8] S.-H. Wang, S.-H. Tai, T. Chiang , "A Low-Power and Bandwidth-Efficient Motion Estimation IP Core Design Using Binary Search", IEEE TCSVT, vol.19, no.5, pp.760-765, May 2009

[9] L.-F. Ding, et al. , "A 212 MPixels/s 4096x2160p Multiview Video Encoder Chip for 3D/Quad Full HDTV Applications", IEEE Journal of Solid-State Circuits, vol.45, no.1, pp.46-58, Jan. 2010

[10] X. Xu, Y. He , "Fast disparity motion estimation in MVC based on range prediction," IEE ICIP, pp.2000-2003, 2008.

[11] H. Shim, C.-M. Kyung , "Selective Search Area Reuse Algorithm for Low External Memory Access Motion Estimation", IEEE TCSVT, vol.19, no.7, pp.1044-1050, July 2009.

[12] S. Saponara, L. Fanucci, "Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems", IEE Computers and Digital Techniques , vol.151, no.1, pp. 51- 59, 2004.

[13] T.-C. Chen, et al. , "Fast Algorithm and Architecture Design of Low-Power Integer Motion Estimation for H.264/AVC", IEEE TCSVT, vol.17, no.5, pp.568-577, May 2007.

[14] P.-K. Tsung et al., "Cache-based integer motion/disparity estimation for quad-HD H.264/AVC and HD multiview video coding", IEEE ICASSP, pp.2013-2016, 2009.

[15] C.-Y. Tsai, et al., "Low Power Cache Algorithm and Architecture Design for Fast Motion Estimation in H.264/AVC Encoder System," IEEE ICASSP, vol.2, pp.II-97-II-100, 2007.

[16] T. Tuan, et al., “A 90nm Low-Power FPGA for Battery-Powered Applications”, ACM FPGA, pp. 3-11, 2006.

[17] S. Yang, W. Wolf, N.Vijaykrishnan, “Power and performance analysis of motion estimation based on hardware and software realizations”, IEEE Transactions on Computers, vol. 54, no. 6, pp. 714-726, 2005.

[18] J. Yang et al., "Multiview video coding based on rectified epipolar lines", ICICS, pp.1-5, 2009.

[19] JMVC 6.0,” garcon.ient.rwthaachen.de, Sep. 2009.

[20] M. Shafique, L. Bauer, J. Henkel, "enBudget: A Run-Time Adaptive Predictive Energy-Budgeting Scheme for Energy-Aware Motion Estimation in H.264/MPEG-4 AVC Video Encoder," IEEE DATE, 2010.

[21] M. Shafique, B. Molkenthin, J. Henkel, "An HVS-based Adaptive Computational Complexity Reduction Scheme for H.264/AVC Video Encoder using Prognostic Early Mode Exclusion," IEEE DATE, 2010.

[22] B. Zatt, M. Shafique, S. Bampi, J. Henkel, "Multi-Level Pipelined Parallel Hardware Architecture for High Throughput Motion and Disparity Estimation in Multiview Video Coding," IEEE DATE, 2011.

Memory Efficient FPGA Implementation of Motion and Disparity Estimation for the Multiview Video Coding

Felipe Sampaio, Bruno Zatt, Sergio Bampi
Informatics Institute – PPGC – PGMicro
Federal University of Rio Grande do Sul (UFRGS)
{fmsampaio, bzatt, bampi}@inf.ufrgs.br

Luciano Agostini
CDTec – PPGC – GACI
Federal University of Pelotas (UFPel)
agostini@inf.ufpel.edu.br

***Abstract* — This paper presents a high throughput and low off-chip memory bandwidth Motion and Disparity Estimation architecture targeting the Multiview Video Coding requirements. The ME and DE modules are the critical paths in the multiview encoding process, corresponding to up to 80% of the encoding time. Besides, these two modules are responsible for more than 70% of the off-chip memory accesses. The goal of this work is to design a hardware architecture that deals with these two constraints. The design space exploration points the best balance between area and throughput. Besides, the Memory Hierarchy allows a reduction of 87% for memory accesses when compared to a solution without memory management. The synthesis results for the FPGA implementation show that the ME/DE architecture is able to process up to 5-view HD 1080p multiview videos in real time in a typical prediction structure with 2 reference frames (temporal and disparity neighbors). When compared to related works, this work presents the best efficiency in terms of off-chip memory access and maximum throughput at this data input.**

Keywords – multiview video coding, memory aware, VLSI design, FPGA implementation

I. INTRODUCTION

Besides the popular HD Televisions, there are several other high realistic applications that have been become popular in the last years. The 3DTV, for example, aims to provide accurate depth perception [1]. In the Free Viewpoint Televisions it is possible to interactively change the viewpoint of a scene [2]. The Wide Television aims to group side-by-side cameras in order to compose a wide view of a scene [3].

To handle with these high realistic applications, the video capture system must generate the so called multiview videos. Multiview videos are the composing of several cameras (views) that are observing the same scene from different viewpoints. The Multiview Video Coding (MVC) is the H.264/AVC extension that deals with multiple views redundancies [4]. Besides the temporal and the spatial redundancies which have already been explored by the traditional monoview encoders, the MVC encoders aim to increase the compression rates by 2 times exploring the inter-view redundancies [5]. Fig. 1 shows a typical prediction structure in the MVC encoders [2].

The Motion Estimation (ME) is part of the Inter Frame Prediction and is responsible to exploit the temporal redundancies, i. e., objects that appear in two or more consecutive frames. The Disparity Estimation (DE), part of the Inter View Prediction MVC innovation, has the goal of reducing the disparity redundancy that is inserted by the multiple scene views. The horizontal arrows in Fig 1 represent ME dependences and vertical arrows are DE operations that are needed to be performed.

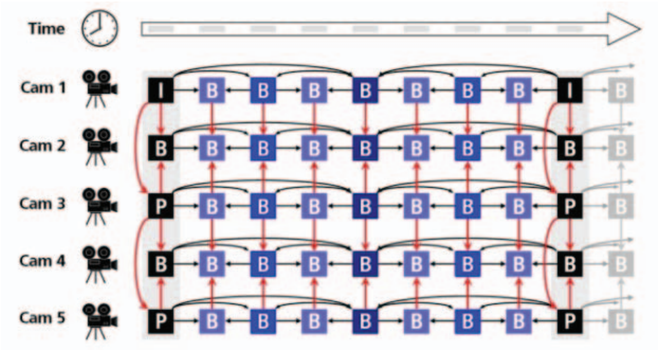


Figure 1. Typical MVC Coding Structure.

The MVC Standard defines that the frame is divided in blocks with 16x16 pixels which are called macroblocks (MB). The MVC ME/DE search is based on a further division of this MB in variable-sized blocks (from 4x4 up to 16x16 pixels), called as current block. The block is searched in a delimited Search Window (SW) of one or more reference temporal or disparity neighbor frames and it is guided by a search algorithm. This work considers the Full Search (FS) algorithm, which performs all possible comparisons. The search is performed by a block matching approach by using some similarity metric between the current block and the candidate block. The most widely used metric is the SAD (Sum of Absolute Differences) [6].

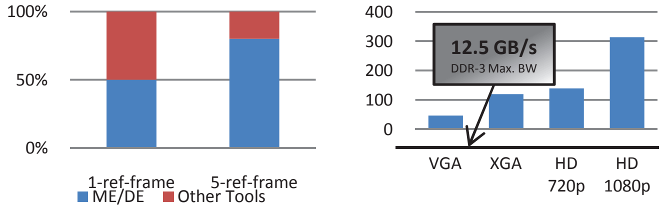
Several issues must be considered in a design of MVC codecs, like the target throughput that is required to achieve real time processing (24~30 frames per second). Besides, another problem is related to how to delivery all the necessary data maintaining the desired throughput. In other words, the memory bandwidth must be efficiently used. In MVC processing, the critical modules in terms of complexity and memory bandwidth are the Motion and Disparity Estimation. These modules are the core of the Inter View and Inter Frame Predictions and are responsible for the highest gains in compression among all coding tools [5].

Fig. 2 shows some experimental results using the JMVC 8.0 reference software [4]. All these results consider the test

978-1-4673-0186-2/12/$31.00 ©2012 IEEE

conditions recommended by the JMVC standardization, normalized in according with [7].

As presented in Fig. 2(a), the ME/DE processing represents 80% of the total encoding time, when 5 reference frames are used as reference. It means that the optimization effort focusing in these modules will result in significant overall execution time reduction. Besides, Fig. 2(b) shows the required memory BW necessary to process only the ME/DE steps in a standard MVC configuration. Then, it is possible to conclude that, even when low resolution videos are processed (VGA for example), the ME/DE data access rate surpasses the maximum data transfer rate of the latest DDR-3 memory technology [8].



(a) (b)

Figure 2. ME/DE (a) encoding time and (b) off-chip bandwidth (in GB/s) in the JMVC reference software.

There are works, like [9], that implement dedicated hardware architecture for Multiple Reference Frames ME targeting the H.264/AVC and exploit parallelism and other techniques in order to achieve high throughput rates. The ME/DE design for MVC is treated in [10] and [11]. These works try to explore some useful multiview characteristics to simplify the hardware design and achieve high processing rates. Outside the architectural design focus, some works propose and discuss data reuse techniques to reduce the off-chip memory bandwidth. There are two approaches: (a) monoview based [12] and [13], and multiview based [14].

This work focuses on two main goals: (1) high throughput and (2) low off-chip memory bandwidth. The high throughput is achieved by exploring the FS inherent parallelism using a column based approach. Besides, the off-chip bandwidth was optimized using some well-known literature schemes (Level D data reuse) [12] and additional frame scheduling processing order (SWCS - Search Window Centric Scheduling) [13].

The designed architecture is able to process two block searches in an interlaced way (ME and DE processing). It is based on a SWCS frame schedule approach and performs the FS of two current blocks using the same Search Window that is managed by a Memory Hierarchy.

The architecture design was focused on a FPGA device. The two on-chip memories were designed to be mapped on Block RAMs, available in all recent FPGA families. Besides, the synthesis results and comparisons consider the architecture implementation on a Xilinx Virtex 5 xc5vlx30 FPGA device [15].

The paper is organized as follows: Section 2 shows an overview of the state-of-art ME/DE hardware architectures; Section 3 explains all the architecture issues: memory organization, processing unities and buffers; Section 4 discusses the synthesis results and presents a comparison with the state-of-art works; finally, Section 5 concludes this work.

II. Related Works

The related works present architectural solutions for Multiple Reference Frames ME targeting monoview video encoding. For comparisons, it is considered the work [9] that proposes a multilevel data reuse scheme target Multiple Reference Frame ME. It is called Multilevel C+. Besides, the work also applies the strategy in a memory aware solution for ME using 4 reference frames.

In [10], the goal is to implement a Fast ME/DE Architecture in addition with other ME/DE architecture. Then, the control unit can decide, based on heuristic metrics, if the Fast ME/DE is used. The goal is to reduce the execution time at each Fast ME/DE execution, even admitting some quality loses. The work [11] presents a cache-based ME/DE algorithm and its VLSI design. The goal is to admit some quality and compression losses in order to save computation and off-chip memory bandwidth.

Data reuse scheme has been target of many works in literature. As they are not restricted to a specific video coding standard, the ideas can be applied for other domains. The works [12] e [13] present consolidated data reuse schemes in candidate block level and search window level. They are called Level A-D and Level C+.

Targeting the MVC standard, the work [14] performs an analysis of required bandwidth and points the challenges for data reuse schemes when encoding multiview videos.

III. ME/DE Architecture Issues

The main goal of this work is to ally high throughput (required for real time processing) with reduced off-chip memory bandwidth. It means that the processing datapath must take advantage of each memory access and performs all possible operations while the data is not discarded.

Some design decisions were assumed based on some evaluations done with the H.264/AVC reference software using HD 1080p videos. When all available block sizes are freely used in the ME, 58% of the chosen blocks are 16x16. When only 16x16 blocks are allowed in the ME process, the PSNR degradation is lower than 0.05 dB using quarter pixel and lower than 0.3 dB if even the quarter pixel is disabled. Since this work is focused in HD 1080p videos, only the 16x16 block size is supported by the designed architecture. This block size is also compliant with the next generation video coding standard [16], where the lowest block sizes are being discarded.

Despite the similarities in the ME and DE processes, there are a few important aspects that differ these two steps. Considering that the ME performs the block match search in previously coded frames in the same view, the search engine will probably find the best match near to the current block position inside the reference frames. It is explained by the high frame exhibition rates that are commonly used (24~30 frames per second). However, when the disparity is considered (DE target) the same assumption is not true. The displacement of the same object in two different views is determined by the camera disparity. This way, the DE effort will depend on the disparity attributes of the target multiview video. The JMVC software deals with this constraint by using large Search Windows (bigger than ± 96 pixels in each direction).

Some important works in literature, like [17], assume an external value to determine the start point for DE searches based on the already know camera disparity of the multiview video. This information is expressed by a vector that is commonly referred as GDV (Global Disparity Vector). This work assumes this information to improve the coding quality results and to reduce the computational complexity. With the GDV usage, it is possible to reduce the Search Window dimensions, mainly for DE searches, because it points for the region where the best match will most probably be. Then, the Search Window was sized as [-8,8) in this work, using the GDV. This means that it bounds the current block in 8 pixels at each direction. Then, the total Search Window dimension is 32x32 samples (256 integer candidate blocks). The Search Window is exhaustive scanned using the FS algorithm.

Fig. 3 presents the overall block diagram of the designed ME/DE architecture with its main modules.

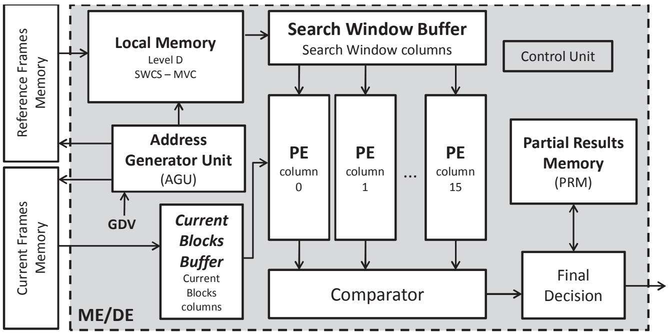


Figure 3. ME/DE Architecture Block Diagram.

The architecture is composed by three main parts: (1) the Memory Hierarchy that implements the Level D - SWCS data reuse scheme, (2) the datapath that synchronizes and processes all data flow and (3) the control modules (AGU and Control Unit in Fig. 3). Further details are described in the next sections.

A. Datapath Architecture

The datapath is composed by a Processing Element Array (PE Array) which is formed by sixteen SAD tree calculator that are able to calculate the partial SAD for one column of 16 pixels. Also, the PE stores the partial SAD in accumulator structures. After 16 clock cycles, the PE deliveries one complete SAD calculation for one candidate block. Each PE was allocated to process a specific column of blocks in the Search Window. Fig. 4 presents this allocation.

In order to synchronize the correct data flow of the Search Window and the Current Blocks, two buffers were designed and sized to store the required information until they are not necessary anymore. The Final Comparator joint all SAD results of each PE column and deliveries the best one to the Final Decision.

B. Memory Hierarchy

The Memory Hierarchy is composed by two on-chip memories. The Local Memory (as shown in Fig. 3) is responsible to store the Search Window samples that are currently scanned by the ME/DE engine. This memory is organized as a circular buffer and it was implemented to support the Level D data reuse strategy. Fig. 5 explains the Level D data reuse scheme.

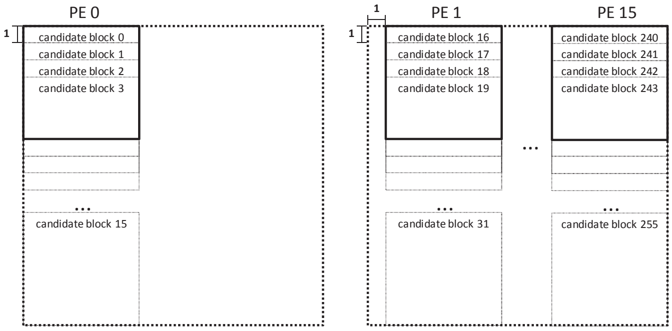


Figure 4. Candidate Blocks Allocation per PE.

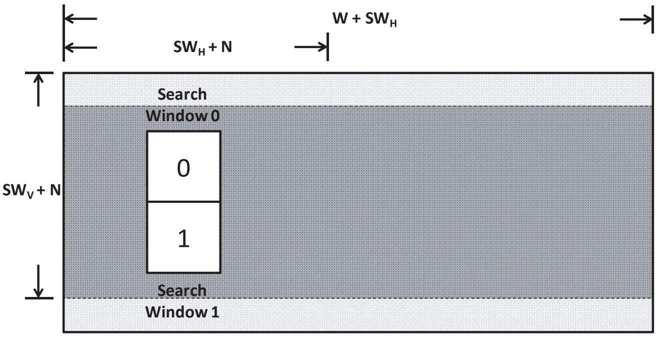


Figure 5. Level D Data Reuse.

When the ME/DE is executed for adjacent current blocks, their Search Windows share a region of the reference frame where both searches must access. This overlapped area is showed in Fig. 5 (the dark gray area). These samples must be accessed a few times in successive ME/DE executions. It implies in redundant memory accesses. The Level D strategy stores locally all frame width. Since the MBs are processed in the raster scan order, when the last MB in a line is finally processed by the ME/DE and the first MB of the next line is initialized, just a few samples (the light gray area in Fig. 5) must be read from the off-chip memory.

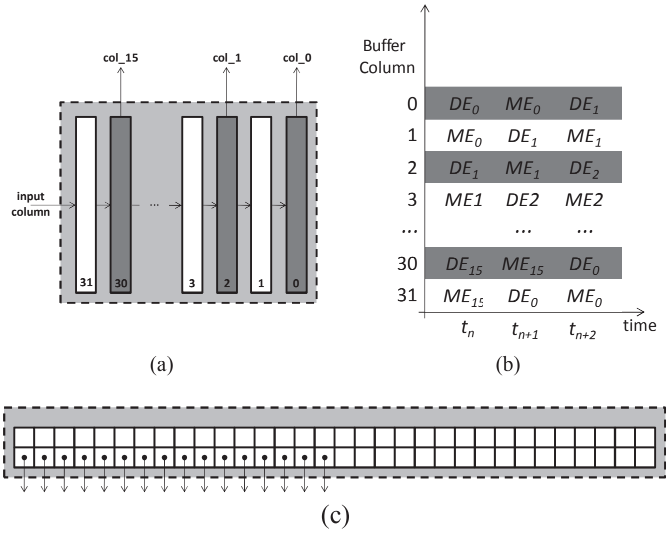
Equation (1) presents a generic calculation that expresses the Local Memory size, where W is the frame width dimension, SW_H and SW_V are the horizontal and vertical Search Window size. The $sample_{bits}$ is the stored samples bitwidth which is used as 8 bits (1 byte) in this work.

$$LevelD_{size} = (SW_H + W) \times SW_V \times sample_{bits} \qquad (1)$$

The Local Memory output is connected to the Search Window Buffer and, at each clock cycle, two samples are accessed and organized in the buffer to be correctly passed to the PE array. In the initialization, the Local Memory requires an initial cycle overhead to be filled with the necessary samples for the first column of the Search Window. However, while the

PE array is calculating the partial SAD results, the Local Memory is written with new samples of the next Search Window columns. This way, the only miss penalty occurs in the beginning of the frame. After, all samples are locally stored when required.

The Current Block Buffer organization (Fig. 6(a) and 6(b)) intercalates ME and DE current block columns. This way, the ME and DE columns are passed to the PE array in an intercalated way. Search Window Buffer organization allows the storage of two Search Window columns of 32 pixels (as presented in Fig. 6(c)).

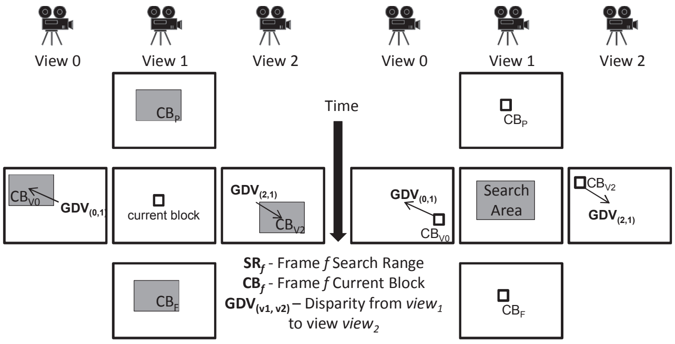


(a) (b)

(c)

Figure 6. Current Blocks Buffer (a) organization and (b) data flow and (c) Search Window Buffer diagram.

Besides, the SWCS scheduling was used to improve the off-chip memory bandwidth reduction. The Fig. 7 presents a representation of this schedule strategy considering the MVC scenario.



(a) (b)

Figure 7. Different Frame-Level Scheduling: (a) CBCS and (b) SWCS.

The SWCS (Search Window Centric Scheduling) is an alternative way to view the video coding order. Instead of considering the current block as the center of the coding process (traditional CBCS schedule – Current Block Centric Scheduling), now the target is to scan one specific reference frame and perform all the ME and DE searches with this Search Window.

The Partial Results Memory (PRM) is required by the out-of-order frame processing required by the SWCS data reuse frame scheduling. Its function is to temporally store the last SAD result of a given block until they are finally decided. When it is decided, then the memory position can be erased and it is available for a partial result of other non-decided block. For each block, the PRM must be able to store: (a) the best SAD distortion information found until that moment, (b) its related motion (or disparity) vector and (c) its reference frame index. The generic formula to calculate the PRM size is expressed in Equation (2), where *W* and *H* are the frame width and height resolutions, *N* is the block size dimension, SAD_{bits}, MV_{bits} and IDx_{bits} are the dynamic range of the SAD, motion vector and reference frame index of each partial decision, respectively. The $frame_{range}$ is the frame range that is needed to be saved until the Final Decision had all information to decide.

$$PRM_{size} = (SAD_{bits} + MV_{bits} + IDx_{bits}) \times \frac{W \times H}{N^2} \times Frame_{range} \qquad (2)$$

C. Pipeline Schedule

The architectural design combines two main ideas: (1) column based Search Window scan and (2) SWCS frame scheduling.

One goal of this work is to efficiently use the off-chip memory bandwidth. Then, the processing order of the overall architecture was defined in order to achieve this requirement. Fig. 8 presents a simple scenario with the configurations that were used in the architectural design.

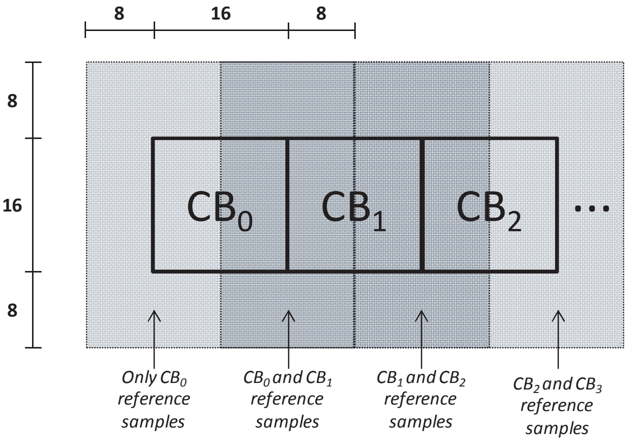

8
16
8
8
16
8
CB_0
CB_1
CB_2
...
Only CB_0 reference samples
CB_0 and CB_1 reference samples
CB_1 and CB_2 reference samples
CB_2 and CB_3 reference samples

Figure 8. Current blocks and Search Windows typical scenario.

The pipeline flow is guided by the Search Window columns. At the first moment, the first column of the Search Windows for the CB_0 is read from the memory. With these samples in the Search Window Buffer, all the possible partial SAD calculations are performed. When there are not more calculations to be performed with these samples, they are discarded and a new column is accessed. In an advanced stage, when the current Search Window column of samples belongs to two Search Windows (the overlap region between CB_0 and CB_1 Search Windows), the architecture must calculate partial SAD for all necessary data of these two current blocks.

The pipeline schedule considers that each Search Window column will be evaluated with all necessary current block columns of the ME and DE in an interlaced way. Fig. 9 shows this pipeline schedule. For example, in the two first time slots, the first Search Window column is matched with the first columns of ME and DE current blocks. This interlaced processing is repeated for all columns.

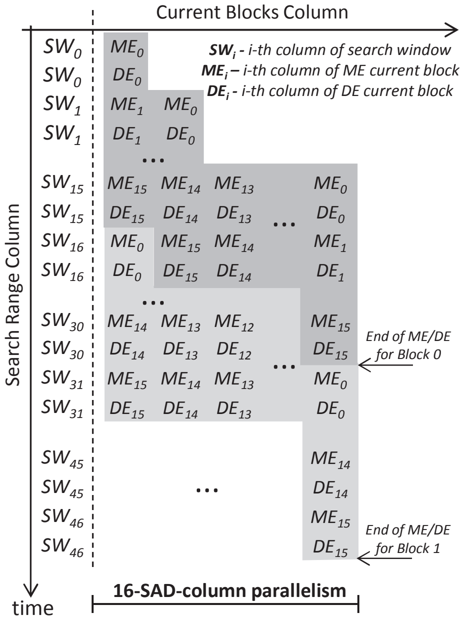


Figure 9. Pipeline and parallelism extraction for the ME/DE architecture design.

Each time slot, with 16 clock cycles, represents the number of cycles available to the PE array to calculate all partial SAD operations. Fig. 9 shows the pipeline filling process and, after the column SW_{15} is accessed, all PE columns are fully processed until the end of the line of the frame, following the raster scan order. Due the overlap Search Windows between two adjacent current blocks, in most of time the PE array will be processing partial SAD of different blocks. Fig. 10 presents the high level timing diagram with this current block processing order.

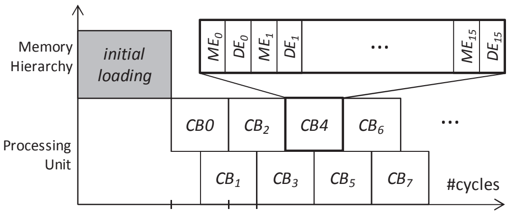


Figure 10. High Level Timing Diagram.

It takes for the ME/DE architecture to perform the FS for the first current blocks (ME and DE current blocks) $32 \times 16 \times 2 = 1024$ clock cycles. Furthermore, because of the pipeline approach, the next two adjacent ME/DE current block searches will be done in $16 \times 16 \times 2 = 512$ cycles. Considering the two target resolutions, XGA and HD 1080p, the architecture spend 33,280 and 61,952 cycles, respectively, to process one entire line of blocks of each ME and DE current frame. This way, a whole reference frame is processed by the ME/DE architecture in 1,597,440 cycles for XGA resolution and in 2,246,400 cycles for HD 1080 videos.

IV. Results

The goal of this work is to design a ME/DE architecture that deals with the MVC performance constraints. Besides, data reuse techniques were employed to reduce the off-chip memory bandwidth. Tab. 1 presents some technology independent metrics that result from the design space exploration. All results are related to a prediction structure using two reference frames (temporal and disparity neighbors).

TABLE I. ME/DE Architecture Results

Specification		XGA (1024x 768)	HD 1080p (1920x1080)
4 views @30fps	Freq. (MHz)	182.8	257.0
	Off-chip BW (Mbytes/s)	91.4	239.3
5 views @30fps	Freq. (MHz)	228.5	321.4
	Off-chip BW (Mbytes/s)	114.6	299.1
6 views @30fps	Freq. (MHz)	274.2	385.6
	Off-chip BW (Mbytes/s)	137.1	358.9
Local Memory (Kbytes)		31	58.1
PRM (Kbytes)		51	143.5

When 4-view processing is the target, the performance requirements for 30 frames per seconds implies in operation frequencies of 182 and 257 MHz for XGA and HD 1080p respectively. In the worst scenario, for 6 views at real time, the real time processing requires 274 and 385 MHz. The architectural design had the premise that the critical path must be of one add/sub operator, in the worst case. This way, frequencies like those required in Tab. 1 are generally surpassed in FPGA implementations. The bandwidth results by using the Level D and SWCS in a jointly way turns feasible the off-chip memory communication. By using these two techniques in the designed memory hierarchy it was possible to decrease the off-chip memory data traffic in 87%, when compared to a solution without data reuse techniques. This way, it is compliant with typical DDR-2 and DDR-3 data transfer rates.

Tab. 2 presents the synthesis results for the FPGA implementation using the ISE 11.0 synthesis tool. The target device was the Xilinx Virtex 5 xc5vlx30 FPGA.

TABLE II. Synthesis Results

	#Slice LUTs	#Slice Registers	#LUT-FF Pairs	Memory Bits	Freq. (MHz)
ME/DE Arch.	19,661 (28%)	12,345 (17%)	9,775 (43%)	1,612.8	369.5

The off-chip memories, the PRM and the Local Memory, were mapped to BRAM (Block RAM) available in the target FPGA device. This way, the Slice Registers were allocated to the designed buffers and the Slice LUTs represent the architecture logical part (the PE array, comparator and control unities). Besides, all carry accelerator structures are useful in the SAD tree operators. The maximum frequency result from the synthesis allows the 8-view processing for a XGA multiview video in real time. For the high definition resolution, the architecture is able to real-time process 5 views of HD

1080p. At this operation frequency, the required bandwidth is up to 300 Mbytes per second.

To the best of our knowledge, there are not FPGA-based architectural implementations targeting low memory access and high throughput for ME/DE processing. So, the comparison will be performed with related ASIC implementations. The works [10] and [11] focus in the MVC in different ways: in [10] a fast ME/DE datapath is inserted in parallel with the usual ME in order to accelerate the MB processing for low motion and low disparity blocks, while the work [11] proposes a new algorithm for ME/DE in order to increase the cache hit of the proposal architecture. Both works admit some PSNR and bitrate losses in order to reduce complexity and memory access rates. Besides, the work [9] was inserted in the comparison by the design similarities, since it also implements a FS architecture for Multiple Reference Frames. In this case, all reference frames are temporal neighbors, since multiview videos are not allowed.

TABLE III. RELATED WORKS COMPARISON

Criteria	**This Work**	**[10]**	**[11]**	**[9]**
Specifications	FS ME/DE MVC	Fast ME/DE MVC	Cache Based MVC	FS AVC
Technology	Virtex 5 FPGA	IBM 65nm LPe LowK	TSMC 90nm	TSMC 180nm
Data Reuse Scheme	Level D SWCS	Level A and Level C	Cache Based	Multilevel C+
Reference Frames	2	4	4	4
Maximum Capability	5 views 1080p @30fps	4 views 1080p @30fps	2 views 1080p @30fps	1 view 720p @56fps
Off-chip Memory BW (Mbytes/s)	300	N.I.	N.I.	204.4
On-chip Memory (Kbytes)	201.6	92.1	7.8	3.96
Efficiency (Throughput/ Off-chip BW)	1,012.5	-	-	132.1

The works [10] and [11] do not clearly inform the off-chip memory bandwidth due the target specifications. This way, complete comparison with these two works was not possible. Besides, as these two works do not use the FS as the target search algorithm, neither a performance evaluation would be fair. It is important to notice that both [10] and [11] aim to achieve high throughput rates by reducing the ME/DE complexity. In [10], it was designed additional processing datapath for Fast ME/DE in order to save computation for low motion/disparity blocks. The work [11] proposes a cache efficient ME/DE algorithm in order to exploit cache hit and misses to direct the search algorithm. However, both works have PSNR and bitrate penalties.

This work applies another design decision: exploit the FS parallelism and regularity in order to increase the throughput without any PSNR drop. The area overhead could not be calculated because of the different target technologies.

The work [9] achieves a better off-chip bandwidth result than our work. However, the achieved throughput is considerably worst. It means that each off-chip memory access is better efficiently used in this work than in [9]. The last row of Tab. 3 presents the efficiency metric that express this trade-off. Considering this aspect, the memory efficiency of this work surpasses [9] in 7.6 times.

V. CONCLUSIONS

This paper presented the FPGA implementation of a ME/DE hardware architecture. The goal was deal with some important issues in the Multiview Video Coding: (a) high computational complexity and (b) high off-chip memory bandwidth. The high throughput was achieved by exploiting the available parallelism in the Full Search algorithm execution. The Memory Hierarchy was designed in order to locally exploit the spatial and temporal data locality. This way, important results in off-chip memory bandwidth reduction were achieved. As main results, the architectural FPGA implementation in a Xilinx Virtex 5 device allows real time processing when using 5-view 1080p HD. The memory access reduction is up to 87% when the Memory Hierarchy is used. The comparison with related works showed that the designed ME/DE architecture has the best efficiency, i. e., the best relation between throughput and required off-chip bandwidth.

REFERENCES

[1] A. Smolic, et al. "Coding Algorithms for 3DTV - A Survey." In: IEEE TCSVT, v. 17, n. 11, pp. 1606-1621, nov. 2007

[2] A. Smolic, et al. "3D Video and Free Viewpoint Video - Technologies, Applications and MPEG Standards." In: 2006 IEEE ICME. pp. 2161 - 2164. 2007

[3] M. Pourazad, et al. "An Efficient Low Random-Access Delay Panorama-Based Multiview Video Coding Scheme". In: IEEE ICIP. Cairo, p. 2945-2948, 2009.

[4] Joint Draft 8.0 on Multiview Video Coding, JVT-AB204, 2008.

[5] P. Merkle, et al. "Efficient Prediction Structures for Multiview Video Coding." In: IEEE TCSVT, v. 17, n. 11, pp. 1461-1473, nov. 2007.

[6] T. Wiegand, et al. "Overview of the H.264/AVC Video Coding Standard". In: IEEE TCSVT, v. 13, n. 7, pp. 560- 576, jul. 2003.

[7] JVT. "Common Test Conditions for MVC". JVT-T207, 2007.

[8] Jedec. "DDR3 SDRAM STANDARD | JEDEC".. Disponivel em: <http://www.jedec.org/standards-documents/docs/jesd-79-3d>, 2011.

[9] M. Grellert et al. "A multilevel data reuse scheme for Motion Estimation and its VLSI design". In: IEEE ISCAS, pp. 583-586, 2011.

[10] B. Zatt et al. "Run-time adaptive energy-aware Motion and Disparity Estimation in Multiview Video Coding". In: 48th DAC, pp.

[11] P.-K. Tsung, et al. "Cache-Based Integer Motion/Disparity Estimation for Quad-HD H.264/AVC and HD Multiview Video Coding". In: ICASSP, . Taipei: pp. 2013-2016, 2009.

[12] J.-C. Tuan, et al. "On the Data Reuse and Memory Bandwidth Analysis for Full-Search Block-Matching VLSI Architecture." In: IEEE TCSVT, v. 12, n. 1, p. 61-72, jan. 2002.

[13] C.-Y. Chen, et al. "Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders." In: TCSVT, v. 16, n. 4, p. 553-558, april. 2006.

[14] P.-K. Tsung, et al. "System Bandwidth Analysis of Multiview Video Coding with Precedence Constraint". IEEE ISCAS p. 1001-1004, 2007.

[15] Xilinx. "Xilinx, Inc.". Available in: <www.xilinx.com>, 2011.

[16] JCT. Work. Draft 3 of High-Eff. Video Coding. JCTVC-E603, 2011.

[17] T.-Y. Kuo, et al. "A novel method for global disparity vector estimation in multiview video coding". In: IEEE ISCAS 2009.