Chapter 2
Background and Related Works

In this chapter the basic notions on digital videos, multiview video systems, and the multiview video coding (MVC) standard are presented. The mode decision, motion and disparity estimation, and rate control modules are detailed since they are the main foci of this monograph. Detailed state-of-the-art review is presented considering 3D-video systems, multimedia architectures, energy-efficient algorithms, and architectures for video coding.

2.1 2D/3D Digital Videos

A video is formed by a sequence of frames (or pictures) of a scene captured in a given frame rate providing to the spectator the sense of motion. Usually, the frame rate goes from 15 to 60 frames per second (fps) depending on the application requirements. Each frame is formed by a number of points named picture elements, i.e., pixels. The number of pixels in each frame is called resolution, i.e., the number of horizontal and vertical pixel lines. The typical resolutions also depend on the target application. For instance, mobile devices use to handle relatively lower resolution and lower frame rate sequences if compared to home cinema that targets high resolution and high frame rates.

Different color spaces are used to represent raw and decoded videos; the most usual ones are RGB (red, green, blue) and YUV. Most monitors operate at the RGB space, while most of video coding standards work over the YUV space. The YUV space is composed of three color channels: one luminance (Y) and two chrominance channels (U and V). The main reason for using YUV space for video coding is related to its smaller correlation between color channels, making easier to independently encode each channel. Since the human visual system (HVS) is less sensible to chrominance when compared to luminance, it is possible to reduce the amount of chroma information without affecting the overall perception. The reduction of chroma information is made using color subsampling (also known as pixel

B. Zatt et al., *3D Video Coding for Embedded Devices: Energy Efficient Algorithms and Architectures*, DOI 10.1007/978-1-4614-6759-5_2,

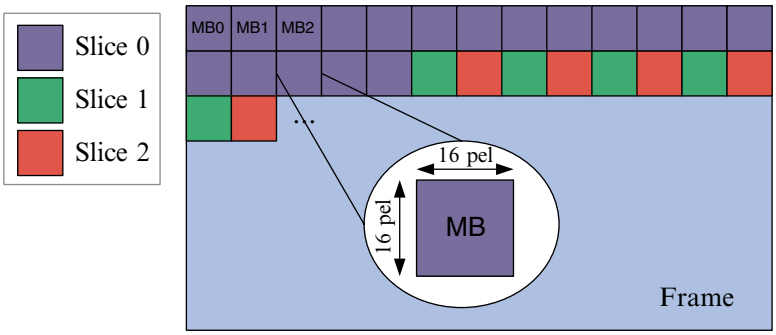

Fig. 2.1 Macroblocks and slices organization

decimation). The most used color subsampling pattern is the YUV 4:2:0 that stores one U and one V sample for each four luminance samples reducing in 50 % the total amount of raw video data.

All current widely used video coding standards are based on block coding. In other words, they divide each frame in pixel blocks to encode the video. These blocks are named macroblocks (MB). In the H.264, the latest video coding standard, the MBs are blocks of 16×16 luma pixels and its associated chroma samples (see Fig. 2.1). A group of MBs is called slice. The slice can be formed by one or more MBs that may be contiguous or not. One frame is formed by one or more slices. In turn, each slice is classified in one of three different types (here the SI and SP slices are not considered): Intra (I), predictive (P), and bi-predictive (B) slices. The example in Fig. 2.1 is composed of three slices: one contiguous (Slice 0) and two noncontiguous slices (Slices 1 and 2). Note, the terminology used here is based on the H.264 standard and is directly applicable to the MVC standard.

For a better comprehension on the different slice types it is necessary to understand the two basic prediction modes used by the state-of-the-art video encoders: intra-frame and inter-frame prediction. The intra-frame prediction only exploits the spatial redundancy by using surrounding pixels to predict the current MB. The inter-frame prediction exploits the temporal redundancy (similarity between different frames) by using areas from other frames, called reference frames, in order to better predict the current MB. Intra (I) macroblocks use the intra-frame prediction, while predictive (P) and bi-predictive (B) macroblocks use the inter-frame prediction. While P macroblocks only use past frames as reference (in coding order), the B macroblocks can use reference frames from past, future, or a combination of both. Intra slices are formed only by I MBs. Predictive (P) slices support I and P macroblocks and bi-predictive (B) slices support I and B macroblocks.

Multiview video sequences are composed of a finite number of single-view video sequences captured from independent cameras in the same 3D scene. Usually these cameras are carefully calibrated, synchronized, and positioned. They are typically aligned in a parallel 1D-array or 2D-array; however, there are systems where the cameras are positioned in arch or cross shapes. The typical spacing between

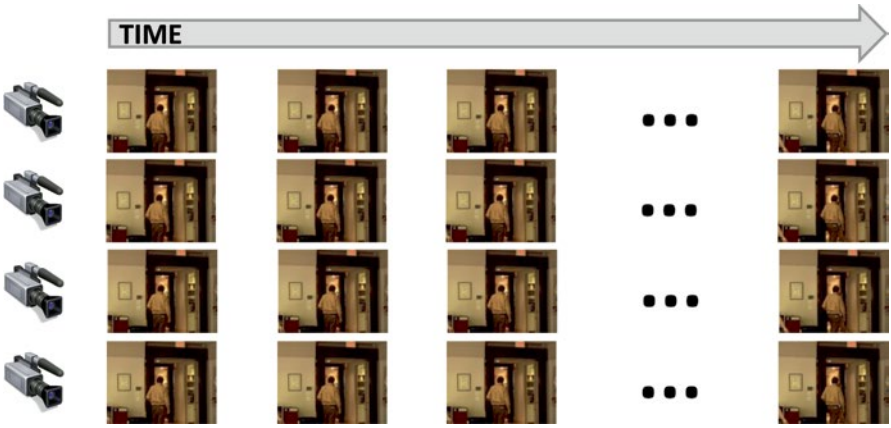

Fig. 2.2 Multiview video sequence

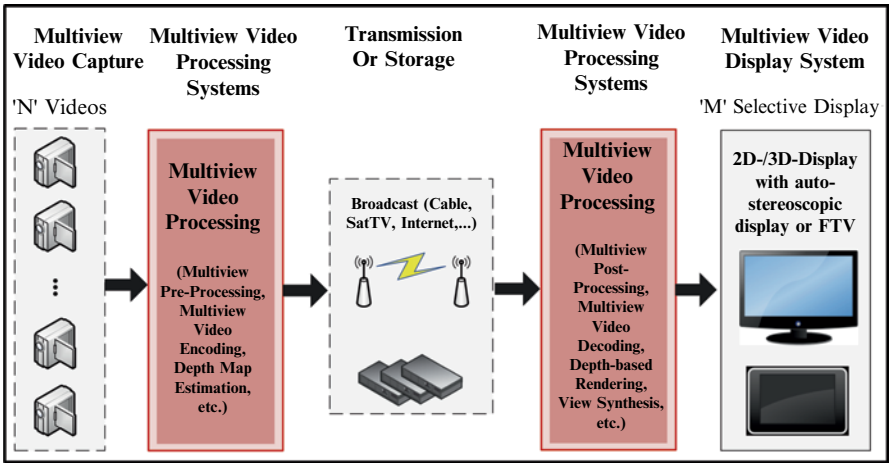

Fig. 2.3 Multiview video capture, (de)coding, transmission, and display system

cameras is 5 cm, 10 cm, or 20 cm for most of the available test sequences. In Fig. 2.2 a multiview video with four views and the captured frames along the time axis is presented. At the video encoding perspective, the MVC, as detailed in Sect. 2.3, extends the concept of inter-frame prediction to inter-view prediction where the correlation between different views is exploited. A deeper discussion regarding the spatial, temporal, and view/disparity correlations is provided in Sect. 2.2.

Figure 2.3 depicts the complete system required to capture, encode, transmit, decode, and display multiview videos. The captured sequence is encoded by an MVC encoder in order to reduce the amount of data to be transmitted. The generated bitstream may be transmitted using broadcast or Internet or stored in media servers

or local storage. At the decoder side the bitstream, or part of it, is decoded and displayed according to the displaying technology available at the receiver end. In a simple single-view display the decoder considers only the base view that is decodable with a regular (H.264/AVC) video decoder. In the case of stereoscopic displays (two views) only two views are decoded and displayed. In free viewpoint television (FTV) systems the user selects the desired viewpoint within the 3D scene and the video decoder selects which views to decode. For multiview displays all views displayed must be decoded along with the reference views used to reconstruct them.

2.2 Multiview Correlation Domains

This section defines the three types of redundancies or correlations present in multiview video sequences in order to provide the background required for a better understanding of the MVC coding tools, detailed in Sect. 2.3, and for the 3D-neighborhood concept presented in Sect. 3.5.1. Here we discuss the correlation at pixel level, i.e., the similarities used to predict the image pixels, and at coding information level, i.e., how neighboring blocks share coding properties such as coding modes, vectors, etc. To have a more general description we present independently the three correlation dimensions (1) spatial correlation, (2) temporal correlation, and (3) view/disparity correlation. Single-view video coding standards are able to exploit (1) and (2), while MVC incorporates (3) to provide improved prediction for multiview videos.

2.2.1 Spatial Domain Correlation

The spatial correlation is the similarity within regions in the same frame. Previous image and video coding standards, such as JPEG2000 and H.263, were already able to exploit this similarity through MB prediction based on neighboring pixels (see Sect. 2.3). Neighboring MBs tend to belong to the same image region and share similar image properties. For this reason, the surrounding pixels typically are good block predictors for the intra-frame prediction process. Exception cases happen in object borders where the image properties may change abruptly. Consider the example in Fig. 2.4, all the MBs in the white background share similar image properties. The same happens for the MBs within one of the objects. The discontinuity occurs when an object border is found leading to increased prediction error. Note that, for simplicity, the spatial correlation is referred as one dimension, but it is actually composed of two dimensions, the width and height of a frame.

On average, the current coding standards are able to efficiently employ the intra-frame prediction for pixel data. However, the correlation of coding side information (coding mode, motion vectors, disparity vectors, etc.) is just superficially exploited. In H.264, a few simple techniques exploit this kind of correlation. The differential

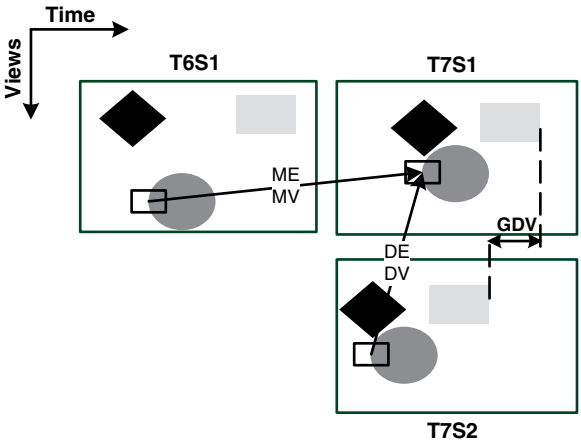

Fig. 2.4 Neighborhood correlation example

coding of intra prediction modes inside a macroblock exploits spatial correlation of coding information. In this technique, the intra coding mode is coded considering the coding mode of the previous block. Another example is the motion vector prediction process that uses the neighboring vectors to predict the current vector. By employing the motion vector prediction, only the differential motion vector needs to be coded and transmitted. These examples show that there is also significant correlation at coding side information level.

2.2.2 Temporal Domain Correlation

The temporal correlation represents the similarities between different frames in the same view of a video sequence. That is, the objects of a given frame are usually present in neighboring temporal frames with a displacement that depends on its motion. Consider the frames T6S1 (view 1, time 6) and T7S1 (view 1, time 7) in Fig. 2.4, the same objects are seen in both frames with a small displacement. Thus, frame T7S1 may be accurately predicted from the reference frame T6S1. The displacement between the two frames is found using the motion estimation (see Sect. 2.3.2). Besides the pixel-level prediction, the coding data are also similar for the same object along the time. In other words, for the same object in distinct time instants the same set of coding modes and motion behavior tend to be employed. The correlation is lost when there is an occlusion or the object moves out of the captured scene.

Analogous to the spatial correlation, there are tools able to exploit the temporal correlation at pixel level, i.e., the motion estimation (Sect. 2.3.2). At coding side information level, an attempt to exploit this correlation was proposed in the H.264 standard by using the temporal direct prediction for motion vectors. This prediction uses the collocated MB (MB sharing the same relative position in the frame) motion vector in order to predict the current one.

2.2.3 *Disparity Domain Correlation*

The disparity is a complete new domain introduced by multiview videos. It refers to the similarities between frames in different views. The similarities or redundancies at pixel level are exploited by the disparity estimation tool (Sect. 2.3.2). However, no tool is able to exploit this correlation at the coding information level.

As depicted in Fig. 2.4, frames T7S1 (view 1, time 7) and T7S2 (view 2, time 7), the same objects are present in the neighboring views displaced by the so-called disparity vector. Since they are the same objects, the same image properties are shared and similar coding information tends to be used in different views. The disparity neighborhood correlation is lost when a given object is out of the area captured by a given camera or there is an object occlusion for a given camera point of view.

In order to obtain an accurate evaluation of the available correlation, we have carried out an extensive analysis of multiview videos. For this analysis we have used different multiview video sequences following the MVC test recommendation by Joint Video Team (JVT) (Su et al. 2006). These sequences have coding structures similar to the one presented in Fig. 2.7. Our analysis, discussed in Sect. 3.5.1, constitutes an in-depth exploration of coding mode distribution, video statistics, motion and disparity vectors, coding mode, and RDCost correlation in the so-called 3D-neighborhood (spatial, temporal, and view neighborhood).

2.3 Multiview Video Coding

Encoding multiview video sequences can be performed using different techniques. The most primitive one is the simulcast approach, where a single-view video coding standard (usually H.264/AVC) is used to encode independently each view. As presented in Fig. 2.5, the simulcast approach considers the intra-frame prediction and inter-frame prediction (a.k.a. motion estimation) exploiting the spatial and temporal redundancies. However, the disparity or inter-view redundancy (i.e., the redundancy between frames of different views) is not considered. The MVC standard uses the inter-view prediction (a.k.a. disparity estimation) to take advantage of the similarities between views from the same scene. The inter-view prediction represented by the red arrows in Fig. 2.5 is responsible for a bitstream reduction of 20–50 % for the same video quality (Merkle et al. 2007). Details on the MVC new tools, coding efficiency, and complexity are discussed along this section.

In a strict definition, the MVC is not a coding standard but an extension of the H.264/AVC or MPEG-4 Part 10 (JVT 2003). The MVC was defined by the JVT in March 2009 (JVT 2008 and JVT 2009b). The JVT is the group of experts formed by the Motion Picture Experts Group (MPEG) from ISO/IEC and the Video Coding Experts Group (VCEG) from ITU-T.

The standard usually works over the YUV (or YCbCr) (Miano 1999) color space that is composed by one luminance channel and two chrominance channels (red and blue chrominance), but other color spaces are supported, such as RGB and YCgCo

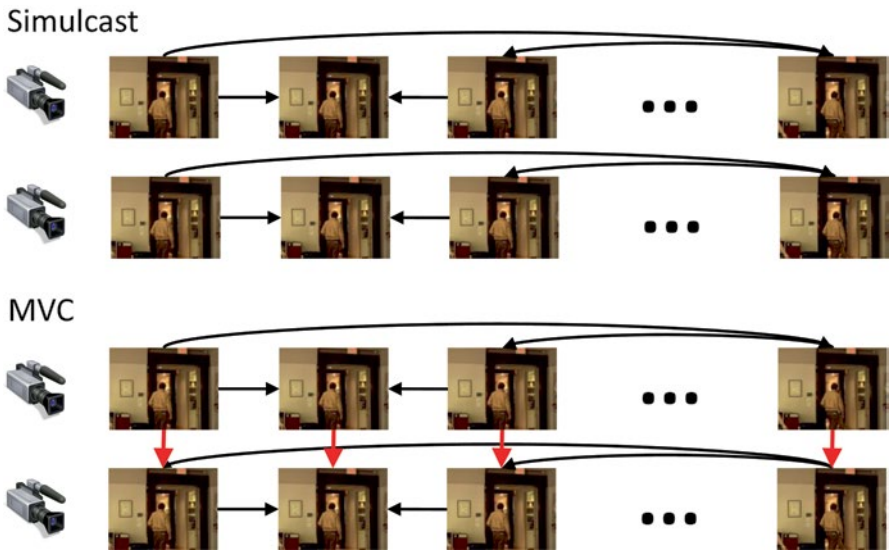

Fig. 2.5 Prediction comparison between simulcast and MVC

(orange and green chrominance). The MVC also supports different subsampling patterns including 4:2:0 (four luminance samples for one sample of each chrominance channel), 4:2:2 (two luminance samples for one sample per chrominance channel), and 4:4:4 (one luminance channel for one sample in each chrominance channel). The supported color space/subsampling and coding tools depend on the profile of video coding operation (JVT 2009a, b).

Originally three profiles were defined in the H.264 standard: Baseline, Main, and Extended. The Baseline profile focuses on video calls and videoconferencing. It supports only I and P slice and the context-adaptive variable length coding (CAVLC) entropy coding method. The Main profile was designed for high-definition displaying and video broadcasting. Besides the tools defined by the Baseline profile, it also includes the support to B slices, interlaced videos, and CABAC entropy coding. The Extended profile targets video streaming on channels with high package loss and defines the SI (Switching I) and SP (Switching P) slices (Richardson 2010). In 2005 the Fidelity Range Extension (FRExt) defined the High profiles: High, High 10, High 4:2:2 and High 4:4:4 targeting high fidelity videos (JVT 2009a, b).

The MVC extension introduced to the standard a new set of CABAC contexts and new supplemental enhancement information (SEI) messages to simplify parallel decoding and the transmission of sequence parameters (JVT 2009a, b). Additionally, the disparity estimation or inter-view prediction was proposed (Merkle et al. 2007). This is the most important innovation in the MVC that allows the exploration of similarities between different views. Its function is to find the best matching for the current macroblock in a reference frame within the reference view. The possible search criteria, search patterns, and objective are similar to the motion

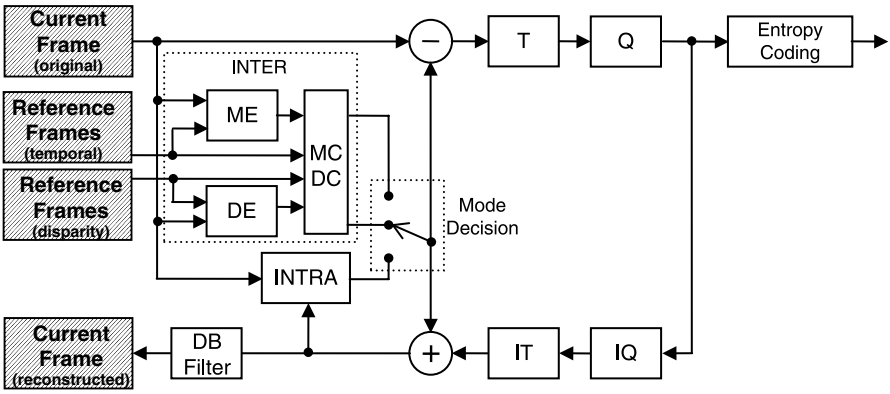

Fig. 2.6 MVC encoder block diagram

estimation. However, the dynamic behavior of the disparity estimation differs significantly with respect to the ME. In the following section, details of the MVC encoding process are presented.

2.3.1 MVC Encoding Process

In Fig. 2.6 the high-level block diagram of the MVC encoding process is presented. As a hybrid coding standard it is composed of three phases: prediction, transforms, and entropy coding. The transform and entropy phases are similar to H.264/AVC, except for the new syntax elements to be encoded by the entropy encoder. The main innovation is in the prediction phase, which incorporates the inter-view prediction tool, the disparity estimation (DE).

The base view, the first one to be encoded, is encoded in compliance to the H.264 standard. Then, the prediction has two options, the intra-frame or the inter-frame prediction. Other views are named dependent views and additionally employ inter-view prediction. The complete encoding process is described in this section, considering the Main profile tools in YUV color space with 4:2:0 subsampling, while further extensions available in the High profiles are omitted for simplicity.

The MVC prediction structure inherits all the possibilities for temporal references and coding orders defined by the H.264. In addition, distinct possibilities of view coding order may be employed. The most used view coding orders are IPP and IBP (Merkle et al. 2007). The prediction structure depicted in Fig. 2.7 employs IBP view coding order using hierarchical bi-prediction (HBP) structure in temporal domain for eight views and group of pictures (GOP) size equals to 8. The set of GOPs for all views are referred in MVC as GGOP (group of groups of pictures). The frames located in the GGOP borders are called anchor frames while all others are the non-anchor frames.

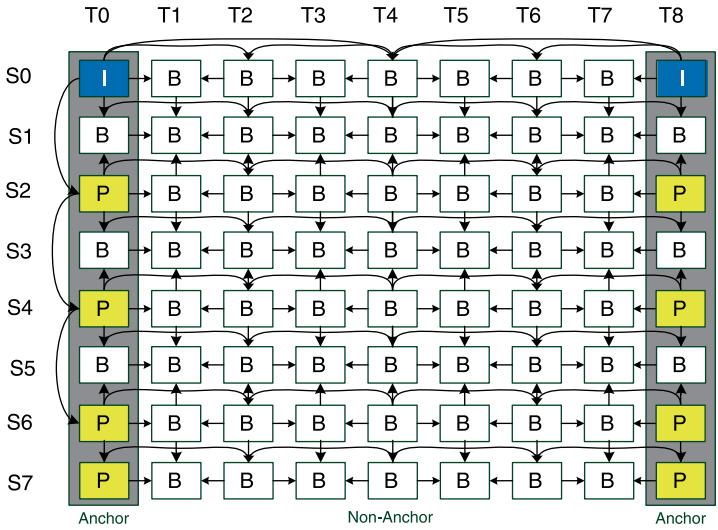

Fig. 2.7 MVC prediction structure example

The intra-frame prediction uses the neighboring pixels within the frame to predict the samples in the current MB. The MVC supports two MBs partitioning sizes for intra-frame prediction. The size 4×4 has nine prediction directions, as presented in Fig. 2.8, where modes 0 and 1 apply a simple copy of the neighboring blocks and modes 3–8 perform a weighted interpolation according to the prediction direction. Mode 2 (DC) replicates the average of the neighboring samples to the entire block. Each one of the 16 blocks inside the MB may use different prediction directions in order to find the best prediction.

The intra-prediction can also be performed using the 16×16 block size. However, in this mode the number of prediction directions is restricted. Figure 2.9 presents the four prediction directions. Modes 0–2 are analogous to the modes 0–2 of the 4×4 block size. The plane mode (3) applies one linear filtering (Richardson 2010) to the neighboring samples resulting in a gradient texture. The 4×4 and 16×16 presented predictions are used for luminance samples. The chrominance prediction uses the same four directions present in 16×16 intra-prediction. The block size depends on the color subsampling; for the 4:2:0 color subsampling, 8×8 chroma blocks are used.

The inter-frame prediction or motion estimation (ME) provides other possibility of prediction. Its function is to perform a search in the past or future previously encoded frames to find the best matching candidate in order to provide a good prediction. The ME features bi-prediction, multiple block sizes, motion vector prediction, ¼ sample motion vector accuracy, weighted prediction, and other tools that help to improve the prediction quality (Richardson 2010), as defined in the H.264/MVC video coding standard and detailed in Sect. 2.3.2.

For the dependent views (all views except the base one), the inter-view prediction or disparity estimation (DE) is also available (Merkle et al. 2007). This MVC

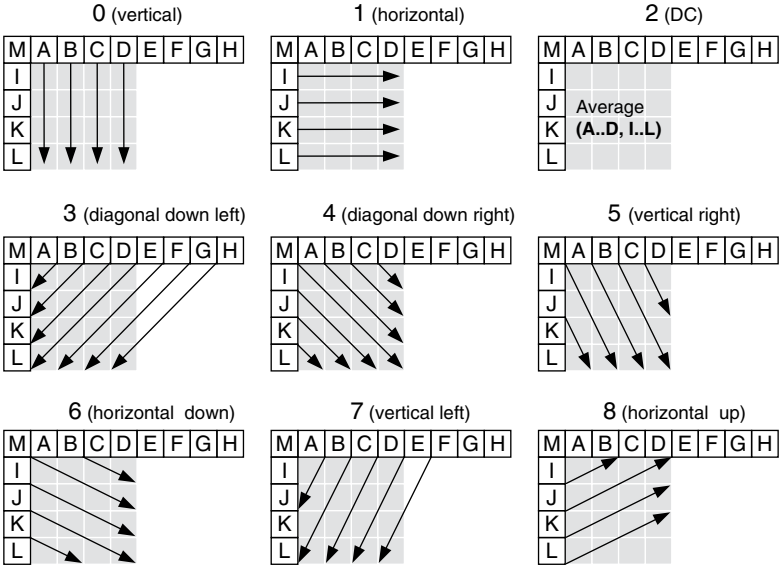

Fig. 2.8 Nine prediction directions for intra-prediction 4×4

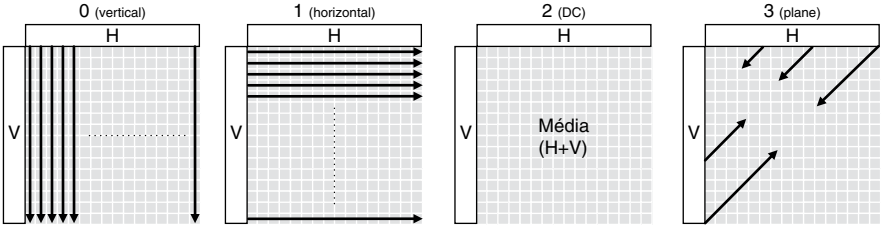

Fig. 2.9 Four prediction directions for intra-prediction 16×16

extension searches for the best matching candidate in the frames belonging to previous encoded views (left, right, up, or down, depending on the cameras arrangement and view prediction structure). All features from ME are supported in DE; more details about these features and how they influence the encoder efficiency and complexity will be discussed in Sects. 2.3.2 and 3.1.1.

The output of the prediction phase is a large set of prediction candidates. Among all different block sizes for intra-prediction, inter-frame prediction, and inter-view prediction, the best prediction mode must be selected by the mode decision (MD) in order to provide the optimal rate–distortion (RD) trade-off (Richardson 2010). The rate is the number of bits required to encode the MB and distortion is the objective video quality measured in peak signal-to-noise ratio (PSNR). To have the optimal solution all modes must be completely encoded, reconstructed, and evaluated according to an RD optimization equation. Therefore, the MD (represented by the selection key in Fig. 2.6) is of key importance since it controls the quality vs.

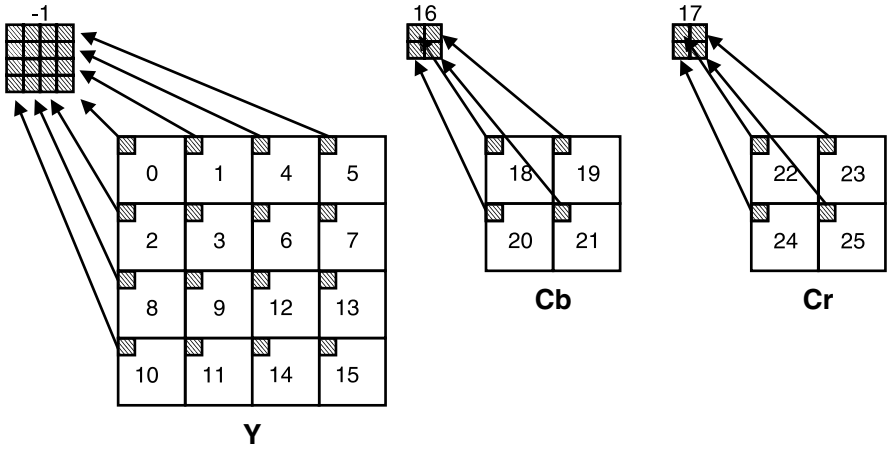

Fig. 2.10 Block processing order in the transform module

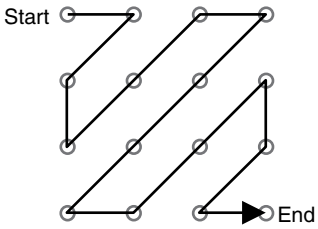

Fig. 2.11 Zigzag scan order for a 4×4 block

efficiency trade-off and the complexity of the encoder. The MD optimization process is discussed in Sect. 2.3.3.

After the prediction phase is completed, the predicted macroblock and the original macroblock are subtracted to generate the image residues. To reduce the energy in a few coefficients the residues are transformed from the space domain to the frequency domain using an integer approximation of the 4×4 2D-DCT transform. If the intra-prediction 16×16 is selected, an additional Hadamard transform is applied after the DCT. In this case, the DC coefficients of each 4×4 block (left upper corner of each block as depicted in Fig. 2.10) compose another 4×4 coefficient block and are submitted to a 4×4 Hadamard transform. The values inside each block in Fig. 2.10 represent the double-Z processing order of the blocks in the transform.

Once the transforms are concluded, each block is quantized to reduce the dynamic range of the coefficients for the entropy coding. In MVC a linear quantization is used. The quantization step is defined by the H.264/MVC standard (Richardson 2010).

Finally, the quantized coefficients are sent to the entropy encoder. Each block is scanned in zigzag order, according to Fig. 2.11, and encoded by one of the two standard entropy encoders: CABAC or CAVLC. The CAVLC use predefined tables

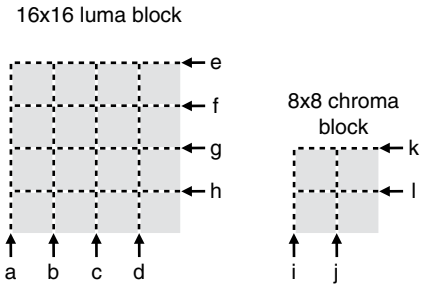

Fig. 2.12 Order of macroblock borders filtering

depending on the syntax element being encoded. The coding method is an evolution of variable length coding to better adapt to multiple contexts. The context-adaptive binary arithmetic coding (CABAC) is a new tool defined by the H.264/AVC standard and implements a novel coding technique able to reduce the bitstream size by about 5–15 % (Wiegand et al. 2003) in comparison to the CAVLC encoder. The tables of probability used in CABAC are updated at bit level and present strong data dependencies. For further information please refer to (JVT 2009a, b; Richardson 2010).

After the entropy coding, the bitstream is assembled and the encoding is complete. However, every macroblock has to be reconstructed to work as reference for further MBs. For that, the inverse quantization and inverse transforms are applied to the quantized coefficients (the same data previously sent to the entropy).

Once the residues are inversely quantized, they are added to the predicted block in order to reconstruct the decoded MB. The reconstruction loop guarantees the consistency between encoder and decoder sides avoiding drifting between encoder and decoder. To reduce the blocky effect (due to different prediction modes) in the reconstructed frames, the standard defines an in-loop deblocking filter (DF). The filtered MBs are used for displaying and to generate the reference for inter-frame and inter-view predictions. Intra-prediction uses unfiltered macroblocks inside a frame. The DF has five filtering strengths and filters the borders of each 4×4 block of the image following the order presented in Fig. 2.12 (Richardson 2010).

2.3.2 Motion and Disparity Estimation

Multiview video sequences are usually captured using a high sample rate, over 30 fps, to improve the motion flow and give the observer a sense of smoother motion. This high frame rate implies in a high redundancy or similarity between neighboring frames in the time axis. As noticed in Fig. 2.13, frames S0T0 and S0T1 are very similar; hence only the differences between them have to be transmitted. The algorithm that exploits these inter-frame similarities is the motion estimation (ME). It searches in the temporal neighboring frames, known as reference frames (see Fig. 2.14), the region that represents the best match for the current block or macroblock. Once the best matching block is found, a vector pointing to that

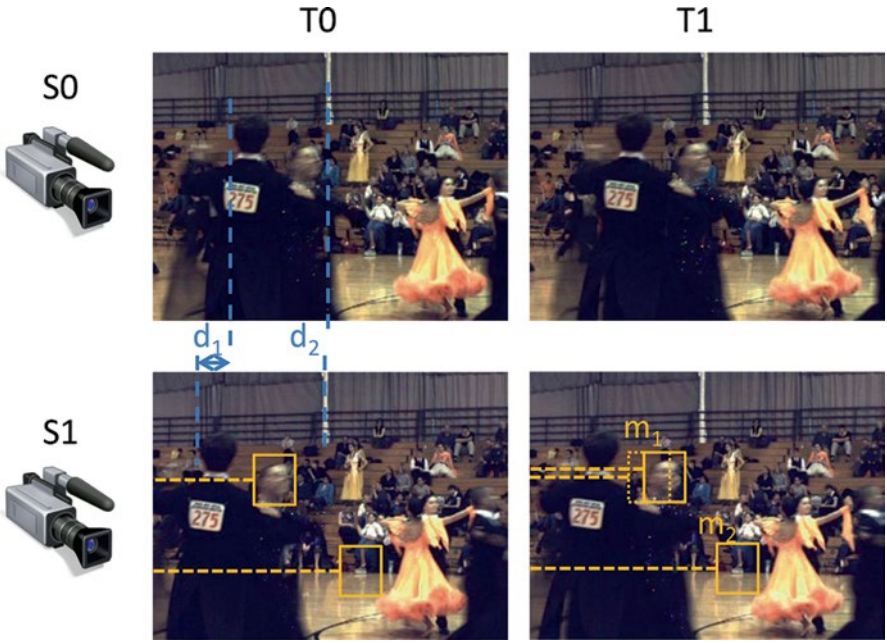

Fig. 2.13 Temporal and disparity similarities

position, the motion vector (MV) in Fig. 2.14, is generated. Consider, for example, a background region (one of the yellow boxes in Fig. 2.13), there is no motion between T0 and T1, so the motion vector m_2 is probably zero. The dancers moving (woman's face in the yellow box) present a displacement along the time; this displacement is represented by m_1. The set of motion vectors of a given frame are called motion field and represent valuable information to understand the motion of an object as time progresses.

The cameras that capture the different sequences in a given 3D scene are located near each other (typically, about 5–20 cm apart) (Su et al. 2006); thus there are many regions that are shared between neighboring cameras. A very high similarity is perceived between neighboring cameras as exemplified in frames S0T0 and S1T0 of Fig. 2.13. The MVC defines the disparity estimation (DE) to exploit the redundancy between different views and minimize the transmission of replicated information multiple times. The approach of the DE is similar to the ME. It searches for the best matching candidate block within frames of the neighboring views. The frame used for search is called reference frame while the view is called reference view, as shown in Fig. 2.14. Once the matching block is found the position is pointed by the so-called disparity vector (DV); see Fig. 2.14. The set of DVs in a frame are referred as disparity field and represent the disparity of the objects between views. While the length of motion vectors (MV) represents the speed an object is moving (or the camera is moving) the disparity vectors denote the displacement of a given object between two views. The disparity depends on the distance between cameras,

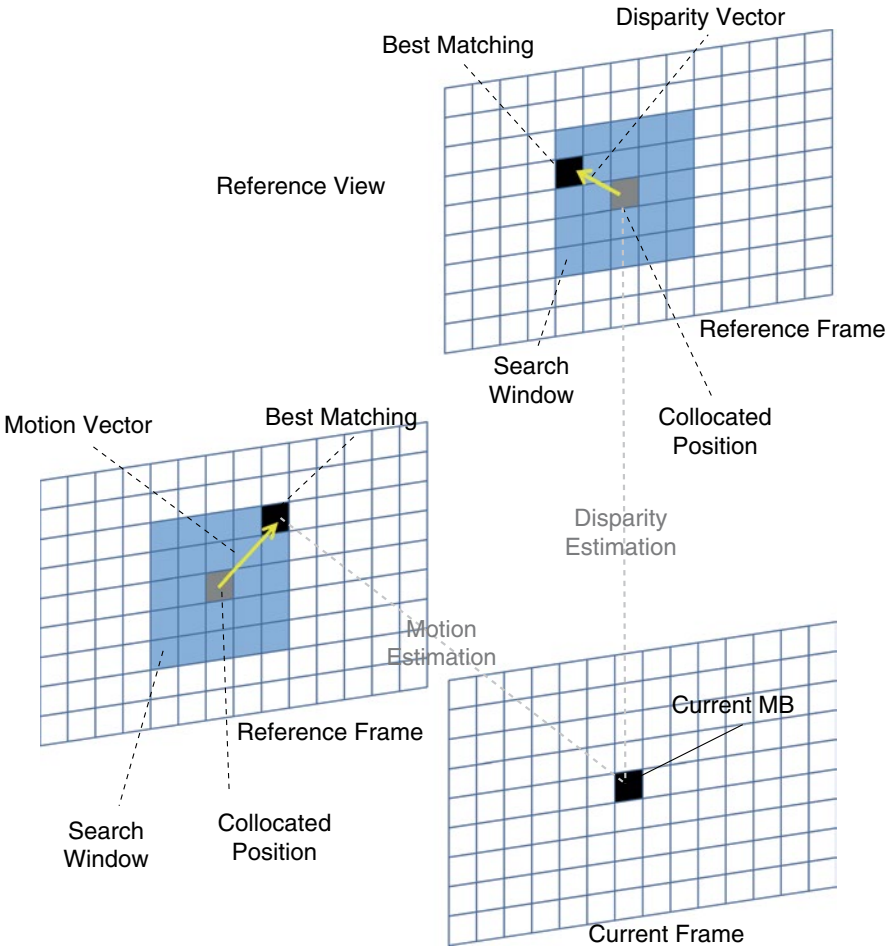

Fig. 2.14 Motion and disparity estimation

and the distance between the camera and the object (Kauff et al. 2007). The closer the object is the larger the displacement or disparity. For instance, in Fig. 2.13, the background presents almost no disparity between S0 and S1 (d_2) while the dancers have a much larger disparity vector (d_1). The average disparity vector between two views considering all objects and background is named global disparity vector (GDV) (Kauff et al. 2007; Han and Lee 2008); see Fig. 2.7.

The ME/DE search is not performed over the complete reference frame but in a region called search window (SW) defined by a search range (SR), as shown in Fig. 2.14, for instance an SR [±16,±16] covers an SW of 33×33 samples. Many search schemes for ME were proposed along the last two decades and their characteristics are well known. The exhaustive search algorithm, the *Full Search* (FS) (Yang 2009), provides the optimal results at the cost of a very high computational

effort. Many fast algorithms focusing on complexity reduction with small quality loss are found such as *Log Search* (JVT 2009a, b), *Diamond Search* (DS) (Kuhn 1999), *Three Step Search* (TSS) (Jing and Chau 2004a), *UMHexagon Search* (Chen et al. 2002) and *Enhanced Predictive Zonal Search* (EPZS) (Tourapis 2002), to list a few. These algorithms are based on multiple search steps oriented by geometric shapes. The most recent schemes also consider the neighboring MBs as predictors to define the search starting point. Using predicted starting point is an evolution compared to the previous search schemes that use the collocated MB as starting point. Recalling, the collocated MB is the macroblock in the reference frame that belongs to the same relative position of the current MB.

Despite the similarity between ME and DE there are behavioral differences that make solutions defined for ME inefficient when applied to DE. For instance, most of the traditional ME fast search patterns perform badly for DE. The reason is that motion vectors are usually located in a relative small length range while disparity vectors usually are much longer. The disparity vectors frequently have 50–100 samples length. For this reason, the recommended search range is at least [±96,±96] for SD resolutions (Xu and He 2008). In this scenario most of the fast algorithms tend to fall in local minima and do not find the optimal candidate. For this reason the JMVC, the reference software for MVC (JVT 2009a, b), implements the *TZ Search* that is more complex in comparison to DS and EPZS, for example, but is still 23× times faster than FS (Yang 2009). The TZ employs predictor centered search start and a larger geometric shape search pattern. It performs well for both ME/DE with negligible or no quality loss in comparison to FS.

However, once the conceptual tasks of ME and DE are similar, the available features are the same and together they represent the most computational and memory intensive tasks in the video encoder; see discussion in Sect. 3.1. For this reason, ME/DE have to be jointly considered in order to propose smart fast algorithms and efficient *architectural solutions for real-time MVC encoding*.

In the following, the motion and disparity estimation features are detailed. Note that all these tools are mandatory at the decoder side depending upon the operation profile but are optional for the encoder.

Bi-prediction: In MVC, two types of MBs employ the ME/DE: Predictive (P), which is coded using inter-frame prediction referencing only past frames and backward views, in display order, or bi-predictive (B), which is coded using reference frames both from past/backward and from future/forward (this is possible due to the out-of-order coding and decoding allowed by the standard). In a B macroblock, each partition can be predicted from one or two reference frames (Sullivan and Wiegand 2005). In case of bi-prediction the final prediction is generated by calculating the average of the prediction from past/backward and future/forward.

The reference frames are stored in two lists: List 0 and List 1. List 0 orders the frames from the past and backward views and List 1 orders the frames from the future and forward views (JVT 2003). Both lists can be ordered using temporal references first or disparity references first. For temporal references first, in List 0 the reference index 0 is the closest past encoded frame. For disparity references

first, the index 0 in List 0 is the closest backward view reference frame. Analogous organization is observer in List 1.

Multiple block sizes: MVC allows ME/DE blocks of several sizes. The 16×16 MB can be segmented in two 16×8, two 8×16, or four 8×8 partitions (JVT 2009a, b). Each 8×8 partition can be segmented in other two 8×4, two 4×8, or four 4×4 sub-partitions. Each partition may point to one reference frame per list (List 0 and List 1) while each sub-partition may use only the frames referenced by the partition that it belongs. Each partition or sub-partition may have a single MV or DV.

Multiple reference frames and reference views: Differently from earlier standards, in MVC the past and future reference frames are not only fixed to the immediate ones. Therefore, to reconstruct one given macroblock, temporally distant frames can be used in the prediction process and this distance is limited only by the size of the decoded picture buffer (DPB) (Sullivan and Wiegand 2005). The reference frames are managed in List 0 and List 1 as previously cited. Analogously, the reference views are not restricted to the closest backward or forward views, any previously encoded views may be used as reference depending on the coding settings.

Quarter-sample motion vector accuracy: In general, the motion of blocks does not match exactly in the integer grid of pixels in a frame, and then fractional-sample motion vector accuracy is used to reach a better match. The MVC (JVT 2003) defines the use of a quarter-sample motion compensation for the reference frame blocks. For luma samples, a six-tap FIR filter is used to interpolate half-samples, and then a simple average of integer and generated half-samples is used to generate the quarter-sample interpolation (JVT 2003). When working with 4:2:0 subsampling, the chroma samples interpolation applies 1/8 sample accuracy.

Weighted prediction: The MVC defines a weighted prediction in the inter-frame coding process to apply a multiplicative weighting factor and an additive offset to each interpolated sample of a given reference frame. For single directional prediction from List 0 or List 1 this tool is defined as presented in Eq. (2.1), where "*x*" is replaced by the list number (0 or 1), "*w*" is the weighting factor, "logWD" is a scaling factor, and "*o*" is the additive offset. *P* represents the interpolated pixels and *P′* the weighted sample. For bi-predictive prediction the weighted prediction is defined as presented in Eq. (2.2):

$$P'(i,j) = ((P_x(i,j) \times w_x + 2^{\log WD - 1}) \gg \log WD) + o_x, \tag{2.1}$$

$$P'(i,j) = ((P_0(i,j) \times w_0 + P_1(i,j) \times w_1 + 2^{\log WD - 1}) \gg (\log WD + 1)) + ((o_0 + o_1 + 1) \gg 1). \tag{2.2}$$

Motion/disparity vector prediction: Exploiting the neighboring blocks correlation, the MVC standard defines that motion vectors and reference indexes (pointer to the reference frame in List 0 or List 1) have to be inferred from the reference index and motion/disparity vectors of neighboring blocks. The inferred vectors are called predicted motion vectors (PMV). Differential motion vectors (MVD) are coded in the bitstream and summed up to the PMVs, obtaining the current motion vector (MV) or disparity vector (DV). The PMVs are normally obtained applying the median to

the spatial neighbor blocks vectors. However, SKIP macroblocks and direct predicted macroblocks (macroblocks with no transmitted residue or motion vectors) are differently processed using the direct spatial or direct temporal predictions. The motion/disparity vector prediction is one example of using the video correlation to predict coding side information, as previously mentioned in Sect. 2.2.

2.3.3 MVC Mode Decision

The MVC provides a big number of options for the macroblocks prediction. Intra-prediction defines two prediction sizes (three in case FREext is considered), 16×16 and 4×4, with four and nine prediction modes, respectively. ME evaluates multiple candidate blocks for seven different block sizes. Additionally, the new disparity estimation adds a set of coding possibilities as large as the motion estimation possibilities.

The mode decision (MD) module is the responsible to deal with this large optimization space. For that it implements an optimization algorithm and defines a cost function called RDCost, the rate–distortion cost (a.k.a. *J* cost). The objective is to evaluate the coding modes and to find the one that minimizes the RDCost to obtain the best coding relation between rate and distortion. Equation (2.3) presents the *J* function where *c* and *r* represent the current original MB and the reconstructed one, *MODE* is the prediction mode used, and *QP* is the quantization parameter. *D* represents the distortion measured after the complete MB reconstruction according to a distortion metric and *R* is the number of bits used to encode the current MB; this number is available once the entropy encoding is completed. λ is the Lagrange Multiplier used to control the rate–distortion trade-off. The Lagrange Multiplier value is not defined by the standard; however, typically it is defined by the Eq. (2.4) and depends upon the QP. To quantify the distortion, different metrics may be used; some examples are sum of absolute differences (SAD), sum of absolute transformed differences (SATD), and sum of square errors (SSE). The SSE is mostly used in the mode decision step since it provides better PSNR results. The reason is that PSNR is calculated using mean square errors (MSE) which is only a division of SSE value, so the SSE is directly related to PSNR. It is important to understand that PSNR is currently the widely most accepted objective video quality metric. However, SAD is widely used in real-time systems due to its light-weight computation:

$$J(c,r,Mode \mid QP) = D(c,r,Mode \mid QP) + \lambda_{Mode} \times R(c,r,Mode \mid QP), \quad (2.3)$$

$$\lambda = 0.85 \times 2^{(QP-12)/3}. \quad (2.4)$$

Although the algorithm to find the mode that minimizes the RDCost is not defined by the standard, the MVC reference software JMVC implements an exhaustive search by completely encoding all possible coding modes and selecting the best mode. It is known as rate–distortion optimized mode decision (RDO-MD) also referred as Full RDO or Exhaustive RDO. The RDO-MD guarantees the optimal MB encoding but drastically increases the encoder computational effort and makes the same approach for real-time MVC encoding unfeasible for the current technology.

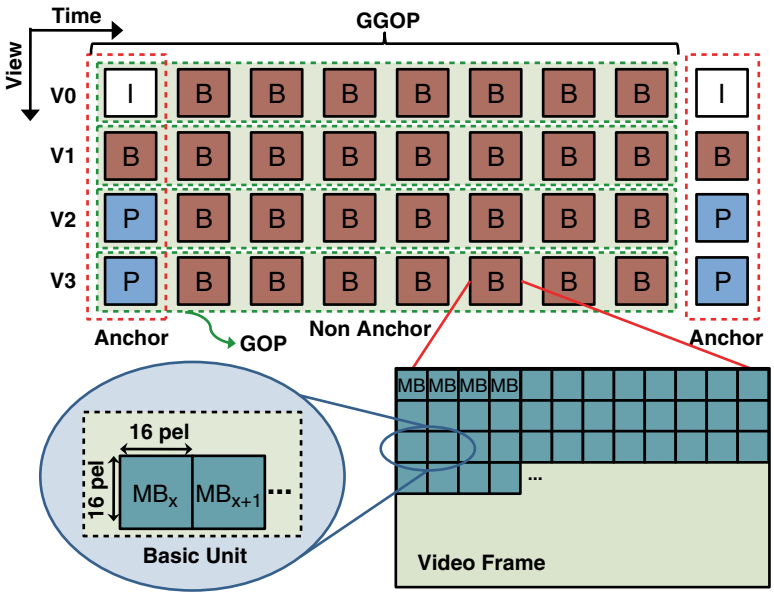

Fig. 2.15 MVC rate control actuation levels

2.3.4 MVC Rate Control

According to, Li et al. (2003) the rate control (RC) is a block of the video encoder that aims to regulate the output-coded bitstream to produce high video quality at a given target bitrate. In the MVC scenario, an efficient RC scheme must be able to provide increased video quality for a given target bitrate with smooth visual quality variation along the time, for different views and within the frames. Most importantly, the RC should keep the bitrate as close as possible to the target bitrate (optimizing the bandwidth usage) while avoiding sudden bitrate variations.

The rate control unit typically controls the quality vs. bitrate through QP adaptation. The bitrate and/or the video distortion metric are predicted using a prediction model. According to the prediction and the target bitrate (amount of bits per second used to encode the video), an adequate QP is selected. As QP grows higher, more residual data are quantized (video details lost) and more quality losses are inserted. The actual generated bitrate and the video quality may be used as feedback for the RC unit in order to update the prediction and QP definition. The QP adaptation may be performed in distinct actuation levels. In general, the RC for MVC can be classified in at least three actuation levels (1) GOP level (group of pictures—set of frames), (2) frame level, and (3) basic unit (BU—set of one or more macroblocks *MB*) level, as shown in Fig. 2.15. It is possible to combine GOP level and frame level, and based on this observation, for simplicity, they are jointly discussed in this monograph.

In the following sections we present the state of the art related to 3D-video systems, MVC encoders, and multimedia processing. Also, a literature overview on the latest low-complexity and energy-efficient solutions focusing on mode decision,

ME/DE and rate control for the MVC standard is presented. An overview on low-power techniques is also provided to give the technical background required for our energy-efficient architectural solutions.

2.4 3D-Video Systems

The advances in video coding techniques targeting multiview videos have been driven by the increasing set of commercial systems employing 3D-video capabilities. These systems range from high-end cinemas and 3DTVs to mobile devices including content suppliers. Wider adoption is expected for the upcoming years with the increase in the available video content through 3D-capable television broadcasters, optical media (Blu-ray Disc Association 2010), popularization of personal 3D camcorders, 3D-video stream services (YouTube 3D 2011; Vimeo 2012), etc. All these commercial systems are, currently, based on stereoscopic videos (two views). An increase in the number of views is expected for the near future (Fujii 2010) to improve the observer freedom and provide a more immersive experience. Some experimental and academic multiview systems are already available or under development to support the next generations of 3D-video systems. In this section we start presenting the most prominent commercial 3D-video systems.

3D-cinema systems are based on three market-leader technologies based on stereoscopic videos (IMAX 2012; RealD 2012; Dolby 2012). The technology employed in IMAX (2012) requires the use of linear polarized glasses to block the light for one eye at a time allowing each eye to see only the frames intended for that eye. Two projectors are used to display 48 fps where each eye is able to effectively see 24 fps in a time-sharing strategy. RealD (2012) is also based on time-sharing between the two eyes; however, the glasses are circular polarized glasses where each glass is polarized in an opposite direction. Also, RealD (2012) requires a single projector able to display 144 fps. Each effective frame for each eye is displayed thrice resulting in effective 24 fps per eye. Finally, Dolby (2012) employs passive glasses with dichroic filters where each view is displayed with a distinct chromatic filter and perceived by a single eye. With this strategy both views are simultaneously displayed allowing the use of regular 24 fps projectors. At the video coding perspective, all these high-end applications support the stereoscopic MVC (The Digital Entertainment Group 2009).

Stimulated by the content available in the 3D Blu-Ray (Blu-ray Disc Association 2010)—optical media that supports the MVC coding standard—the 3D televisions already exceed 10 % of the televisions sold in the United States, in 2011, and this number is expected to reach 37 % of the market share in 2014 (Research and Markets 2010). Other countries are expected to follow this trend. The majority of those 3DTVs are based on stereoscopic displaying and require active shutter or passive polarized glasses to provide the 3D sensation. Many devices employ built-in decoders supporting the MVC standard. Along with the cinema solutions, the 3D televisions are not energy-critical and typically implement only the video decoder side, less complex in relation to the encoder.

Currently, portable devices capable of handling digital video are available everywhere for a reasonable cost. The omnipresence of these gadgets implies a very large amount of data being produced. In this scenario the coding efficiency is a key issue in order to reduce the storage and transmission costs for digital video. Various devices are also capable of real-time 3D-video recording, such as Panasonic (2011), Fujifilm (2011), Sharp (2011), and Sony (2011). Most of them feature two cameras and encode the video sequences independently (simulcast). However, the increase in number of views from 2 up to 4–8 views (Fujii 2010) in order to provide enhanced 3D experience freedom is envisaged for the next 3–5 years. In this scenario it is simple to conclude that the large amount of data generated requires the use of the state-of-the-art MVC standard. The first personal camcorder to fully support stereo MVC was released by Sony in 2011.

Although 3D-capable mobile devices are already available, attending MVC performance and energy constraints remains a big challenge for industry and academia, as discussed in Sect. 3.2. The current multimedia processing systems based on processors, DSPs, and non-MVC-optimized application-specific integrated circuits (ASIC) implementations are not efficient to provide the required throughput with the required energy efficiency while sustaining video quality and coding efficiency. In the following section we present an overview of the main multimedia architectural approaches and solutions in the current state of the art.

2.5 Multimedia Architectures Overview

In this section we present the multimedia processing architectures classified in four main classes: Multimedia Processors/DSPs, Reconfigurable Multimedia Processors, ASIC Multimedia cores, and Heterogeneous Multicore SoCs. On the one hand, ASIC solutions provide the highest performance and energy efficiency at the cost of reduced flexibility limiting the applicability to upcoming video standards. Still, the current lack of MVC-oriented ASIC optimizations prohibits further increase in both performance and energy efficiency. On the other hand, multimedia processors/DSPs allow high flexibility to multiple standards while providing reduced performance and poor energy efficiency if compared to ASICs. Additionally, reconfigurable processors may allow significant increase in performance and flexibility through instruction set architecture (ISA) extensions. The reconfigurable processors, however, present reconfiguration energy issues and are unable to reach the ASIC-like performance and energy efficiency required by the 3D multimedia applications.

2.5.1 Multimedia Processors/DSPs

Aware of multimedia processing characteristics, the Multimedia Processors/DSPs are designed to exploit the parallelism inherent to these applications. Massive multicore architectures are proposed to target task parallelism by supporting multiple

parallel threads. Data-level and instruction-level parallelisms are exploited by employing single instruction multiple data (SIMD) and very large instruction word (VLIW) architectures, respectively. Some proposals are able to implement hybrid parallelism by handling multiple cores with SIMD and/or VLIW instruction sets.

In Abbo et al. (2008), the Xetal-II employs 320 SIMD processing elements with a dedicated 10 Mb on-chip frame memory. It is able to provide 107 GOPS with a 60 W power consumption with instructions designed targeting video analysis applications. A multicore system for video decoding is proposed in Finchelstein et al. (2009) employing a caching mechanism to reduce the memory reads. The work presented in Khailany et al. (2008) describes a processor with 16 parallel lanes where each lane is a 5-ALU VLIW core. At 800 MHz, this solution delivers 512 GOPS (82 pJ/MAC) and guarantees baseline HD1080p H.264 encoding at 30 fps. The multi-streaming SIMD multimedia engine proposed in Chiu and Chou (2010) claims a 3.3–5.5× performance increase compared to MMX architecture (Intel Multimedia Extension) by employing 12 multimedia kernels. These parallel architectures provide a relative high performance but are still far below MVC requirements and the power envelope is out of embedded devices boundaries.

A 2-issue VLIW stream processor is presented in Chien et al. (2008) with throughput for CIF encoding at 30 fps. Stereo processing-oriented optimizations for VLIW processors are presented in Payá-Vayá et al. (2010). The authors claim performance improvements by implementing a new register file access mechanism and disparity functional unit to calculate disparity map. Also, an application-specific instruction processor (ASIP) based on a VLIW DSP architecture is described in Zhang et al. (2009) and delivers increased performance if compared to traditional DSP and SIMD.

2.5.2 Reconfigurable Processors for Video Processing

In Otero et al. (2010) an architectural template for run-time scalable systolic coprocessors is presented. It focuses on run-time adaptation to time-variable tasks or changing system conditions. It exploits replacing and relocation of basic processing elements of the array using FPGAs dynamic reconfiguration. In Beck et al. (2008) is employed a coarse-grained reconfigurable array with a run-time mechanism designed to translate MIPS instruction to be executed in the reconfigurable array. Berekovic et al. (2008) present the mapping of MPEG-2 and H.264/AVC to the ADRES (coarse-grain reconfigurable processor) delivering throughput for real-time CIF decoding at 50 MHz with a 4×4-core array. CRISP, a coarse grain reconfigurable stream processor (Chen and Chien 2008), implements an image processing pipeline reaching 55 fps for HD1080p resolution. Aggressive performance losses are expected for video coding due to increased complexity compared to the implemented image processing algorithms.

In Bauer et al. (2007), the rotating instruction set processing platform (RISPP) is presented bringing more flexibility to extensible processors. It features a special instruction forecasting algorithm able to predict the hotspots and allows to adapt at

run time the different *Molecules* (implementation of the special instructions). This architecture was evaluated using some H.264 processing hotspots (SATD, DCT, etc.) and demonstrated high flexibility to deal with the performance vs. hardware trade-off. This concept was extended in Bauer et al. (2008a) by integrating a special instruction run-time scheduler able to outperform the state of the art in 2.38× for the H.264 application. When integrated to a transmutable embedded processor (Bauer et al. 2008b), the RISPP concept was able to present up to 7.19× speedup in relation to related works for H.264.

Compared to regular processors, reconfigurable processors target to increase the overall performance by adapting, at run time, to distinct applications properties. Also, the adaptivity can be efficiently exploited within the same application. Considering multimedia applications, the performance/energy requirements may vary with the video content, user settings, battery level, etc. It brings a big optimization potential at the system perspective. However, when considering a single application for a given description, in this case real-time encoding for MVC HD1080p, the profit of this adaptive behavior is not perceived. Moreover, in this scenario, the energy and time costs for reconfiguration pose additional difficulties in terms of throughput and energy efficiency if compared to processors, DSPs, and ASIPs.

2.5.3 Application-Specific Integrated Circuits

Multiple ASIC hardware architectures were proposed targeting real-time high-definition (de)coding in accordance to the latest video coding standards trying to reduce the total energy consumption for embedded devices. The architecture presented in Chen et al. (2009c) delivers H.264 encoding for D1 (720×480) resolution with 43.5–67.3 mW consumption. In Lin et al. (2008) an H.264 video encoder able to process HD1080p sequences at 242 mW is presented. Chang et al. (2009) propose a real-time 720p H.264 encoder at 183 mW consumption. It implements a 3-stage pipeline, 8-pixel intra-prediction parallelism, and a parallelized subsampling algorithm. A 59.5 mW AVC/SVC/MVC decoder for up to three-view HD1080p videos is presented in Chuang et al. (2010). The first complete video encoding solution for MVC encoding was presented in Ding et al. (2010). The AVC/MVC 3D/Quad Full HDTV supports three-views HD1080p and consumes 522 mW.

Compared to other approaches, ASIC provides high throughput and energy efficiency. Considering the state-of-the-art IC manufacturing technology, ASIC implementation is the only solution able to encode high-definition MVC for an increased number of views at real time, as shown in the related work overview presented. Still, further optimizations are possible in relation to the presented ASIC-related works. Having Ding et al. (2010a, b) as comparison basis, the future MVC encoding systems will require increased number of views. Moreover, in Ding et al. (2010a, b) single-view-based optimization techniques (such as search window reduction that seriously affects the disparity estimation) are employed leaving a high potential for multiview-aware optimizations.

2.5.4 Heterogeneous Multicore SoCs

Heterogeneous multicore architectures are also proposed targeting multimedia applications. In Kollig et al. (2009), the proposed systems handle an ASIC HW video codec, audio codecs, VLIW processors, MIPS host CPU, DSP, and other HW accelerators. The system proposed by Kondo et al. (2009) is composed of two specific accelerators (video decoder and descriptor), one general accelerator (MX), three RISC CPUs, and caches. Woo et al. (2008) describe a 195 mW mobile multimedia SoC with ARM 9, AVC/MPEG decoder, JPEG codec, fully programmable 3D engine, and multiple peripheral interfaces. The heterogeneous multicores approach is the most used in the current set-top boxes, digital TV decoders, and smart mobile devices. It takes advantage of some degree of flexibility along with the performance of specific accelerators.

The SoCs in current commercial mobile devices such as smartphones and tablets implement heterogeneous multicore SoCs employing processors with SIMD extensions, DSPs, ASIC codecs or hardware accelerators, and programmable embedded GPUs. Qualcomm Snapdragon S4 (Qualcomm Inc. 2011) is composed of up to 4 ARM cores, Hexagon DSPs, video coding hardware accelerators, and the Adreno embedded GPU. Nvidia Tegra 3 (Nvidia Corp. 2012) is based on up to 4 ARM cores and employs dedicated video encoder/decoder and ULP GeForce GPU. Samsung Exynos 4 (Samsung Electronics Co. Ltd. 2012) is composed of quad-core ARM processor, video/image/audio ASIC codecs, and the ARM Mali GPU. Texas Instruments OMAP 5 (Texas Instruments Inc. 2012) employs 2 ARM Cortex-A15, 2 ARM Cortex-M4, DSPs, video/audio accelerators, and the PowerVR GPU. Note, even with efficient ARM processors, SIMD extensions, DSPs, and programmable massively parallel GPUs, the embedded SoCs require ASIC codecs/acceleration units to deliver the throughput and energy efficiency for real-time high-definition video encoding. To deal with multiview videos and attend performance/energy requirements on embedded battery-powered devices, these SoCs will require MVC-oriented optimizations at algorithmic and architectural (including datapath and application-aware units/memory management optimizations) levels.

2.6 Energy-Efficient Architectures for Multimedia Processing

In this section we introduce the state of the art on energy management along with an overview on video memories, energy-efficient techniques, and architectures for multimedia processing. Additionally, the infrastructure to support dynamic voltage scaling (DVS) on SRAM memories and the dynamic power management (DPM) schemes are presented. This technique is extensively used in the literature and in the solutions proposed along this monograph.

2.6.1 Video Memories

On-chip memories are becoming dominant part of current systems, mainly for signal processing systems. In the scope of video coding, the video memory represents the main on-chip memory component responsible for storing frames used as reference to encode other frames. In the current literature are found solutions specific for video/frame memories or generic solutions for any video/image processing tasks. Some of these solutions are described in the following.

The work in Grun et al. (1998) proposes a memory size estimation method for applications containing multidimensional arrays such as video processing. The memory estimation is generated from the application algorithm specification. The paper also addresses the discussion relating parallelism to the memory size. Zhu et al. (2006) present a memory size computation method for multimedia algorithms. The solution uses algebraic techniques and the theory of integral polyhedral to compute exactly the memory size for multimedia algorithms. The authors in Yamaoka et al. (2005) use a triple-mode SRAM to implement an on-chip memory for mobile phone application. The on-chip memory is composed of four SRAM banks that can be managed by a leakage state controller.

In terms of specific video memories, the authors of Shim and Kyung (2009) propose a video memory composed of multiple on-chip memories employing a data reuse to reduce the external memory access. A memory switching method is defined to increase the utilization of on-chip memory. Tsai et al. (2007) present a low-power cache for the reference frame memory of H.264/AVC. This work uses the block translation cache architecture and a search trajectory prediction prefetching scheme. The authors claim a 35 % memory writing power reduction with 67 % memory static power reduction.

2.6.2 SRAM Dynamic Voltage-Scaling Infrastructure

The static energy due to leakage current represents a significant source of the total energy consumption in deep submicron technologies. Also, current integrated circuit footprints are dominated by embedded memories which are typically implemented as SRAM (static random access memory). Therefore, reducing SRAM static consumption is a key challenge to reach overall energy reduction.

The fabrication technology evolution has provided meaningful contribution to leakage reduction by employing high-K oxides (Huff and Gilmer 2004), FinFET transistors (Pei et al. 2002), etc. Along this monograph we assume the use of an on-chip SRAM memory featuring multiple power states with data retention capabilities. The high-level memory organization is presented in Fig. 2.16. An implementation for this memory organization including a picture of the silicon die is demonstrated in Zhang et al. (2005). Still, there is a need to further reduce the leakage at architecture and system levels through techniques such as power gating, DVS, and DPM. In Sects. 2.6.3 and 2.6.4, we present an overview of power/energy management techniques for memories and multimedia systems, respectively.

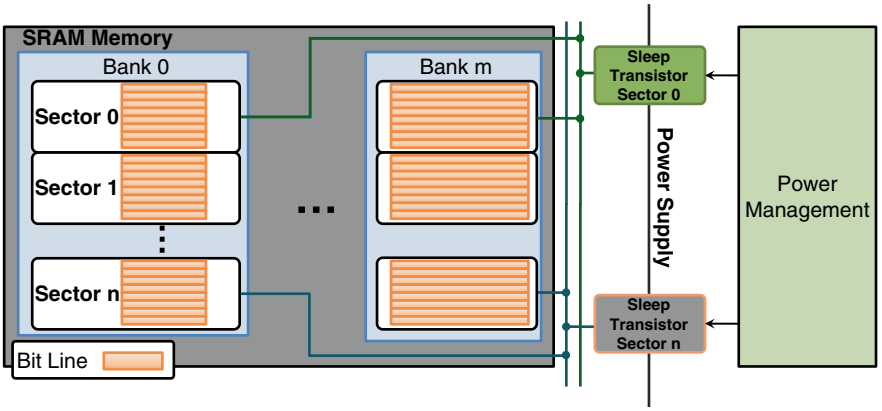

Fig. 2.16 SRAM voltage-scaling infrastructure

2.6.3 Dynamic Power Management for Memories

In order to take advantage of the on-chip memory organizations (such as those presented in Sect. 2.6.1) and multistate SRAM infrastructures (see Sect. 2.6.2) in order to convert these features to actual energy savings, efficient DPM scheme is required. Some DPM solutions in the context of video processing and embedded systems are discussed below.

Panda et al. (1997) present a local memory optimization technique for embedded systems based on memory performance analytical estimation for a given application. The base architecture employs cache and a scratch-pad memories with parameters defined by the proposed technique. In Cong et al. (2009) the memory performance and energy optimization are performed through automatic memory partitioning and scheduling. Firstly, the solution considers the cycle accurate application schedule to meet the memory performance requirements. Secondly, the memory banks are partitioned to reduce dynamic power consumption. The work in Wang and Mishra (2010) implements a dynamic cache reconfiguration technique along with DVS for memory system energy minimization.

Generic techniques for reducing the on-chip SRAM leakage [like (Singh et al. 2007; Agarwal et al. 2006)] propose memories with multiple sleep modes in order to better exploit the leakage vs. wake-up penalty trade-off. State-retentive power gating of register files featuring multiple sleep modes is presented in Roy et al. (2011). In Mondal and Ogrenci Memik (2005) the hardware power gating is controlled by monitoring the underlying hardware. These observation-based techniques may lead to mispredictions, especially in case of sudden variations. The techniques in Liu et al. (2008) and Rajamani et al. (2006) consider application knowledge for a video decoder case study, but they only exploit the knowledge at frame level. These techniques consider longer periods and may not cope with severe variations at the MB level.

In Fukano et al. (2008) a DVS using dual power supply is used to implement a 65 nm SRAM memory employing three operation modes (1) high speed, (2) low power, and (3) sleep mode. In this work the low power and sleep modes are data retentive avoiding data refetching, but it does not support partial DVS for specific sectors of the SRAM. Yamaoka et al. (2004) present a similar solution employing three operation modes while supporting bank-level DVS. It achieves leakage reduction through adapting the virtual supply voltage using PMOS transistors. Finally, the 65-nm SRAM design presented in Zhang et al. (2005) provides more flexibility through adopting multiple power states and fine grain power control. The DVS is controlled at sector level using a custom NMOS sleep transistor to control the virtual ground voltage.

2.6.4 Energy Management for Multimedia Systems

Energy and power management for multimedia systems has been studied in many research works mostly targeting embedded applications. The authors in Cao et al. (2010) employ DVS with five distinct voltage levels. It is controlled using the application-specific knowledge through workload modeling for a wavelet video encoder. In Kamat (2009) a battery level-aware MPEG-4 video encoder with a notification manager and an application-specific controller is presented. Some solutions exploit the energy vs. video quality trade-off at run time to adapt to the system scenario. Ji et al. (2010) partition the input data in distinct profiles used for energy budgeting generating scalable video quality according to the energy scenario. Similar work is presented in Ji et al. (2009) applying game-theory algorithms to control the video encoder. Liang and Ahmad (2009) propose a rate–complexity–distortion model to progressively adjust the H.263+ encoder behavior considering the video content. It employs DVS providing and reaches up to 75 % energy reduction. A power-rate–distortion model (He et al. 2008) is used for energy minimization in video communication devices by exploring energy trade-off between video encoding and wireless communication providing up to 50 % energy reduction. A dynamic quality adjustable H.264 encoder is proposed in Chang et al. (2009). It defines quality states to change the number of coding modes considering the power vs. quality trade-off. The implemented ASIC provides real-time 720p encoding at 183 mW consumption. The proposals summarized in this section are useful at the MVC scenario but lack the MVC-specific knowledge such as workload model, quality states, rate–distortion behavior, etc. Thus, the simple application of these approaches leads to inefficient energy management performance.

Authors in Javed et al. (2011) presented an adaptive pipelined MPSoC for H.264/AVC with a run-time system that exploits the knowledge of macroblock characterization based on their spatial and temporal properties (Shafique et al. 2010; Shafique et al. 2010a) to predict the workload. Based on this knowledge, unused processors are clock-gated or power-gated. These techniques provide limited energy-efficiency in MVC as they cannot exploit the MVC-specific knowledge such as (a) distribution of memory usage at frame and MB levels and (b) memory usage correlation in the 3D-neighborhood.

2.6.5 Energy-Efficient Video Architectures

The work of Shafique et al. (2010) presents an energy budgeting scheme for the H.264 ME. This solution considers the total energy available along with the video properties to dynamically define a search pattern able to deal with the energy vs. quality trade-off. Each frame is classified into one of six energy classes and further classification refinement is performed at MB level. The highest complexity class performs a search composed of three search patterns (Octagon Star, Polygon and Diamond) without samples subsampling. The lowest complexity class employs a Diamond-shaped search using 4:1 subsampling. The highest complexity class requires 17× more energy when compared to the lowest complexity class.

The authors in Chen et al. (2006) evaluated different state-of-the-art data reuse schemes (Level-A, Level-B, Level-C, and Level-D) and proposed a new search window-level data reuse for H.264 ME (Level-C+) in order to reduce the energy consumption related to external memory access and on-chip memory storage. Level-A and Level-B solutions are based on candidate blocks. While Level-A fetches and stores on-chip a single candidate block, Level-B fetches a whole candidate stripe (inside the search window). They require frequent external memory access and only fit with regular search patterns which is not the case for state-of-the-art ME/DE algorithms. Level-C and Level-D follow the same logic but at search window level. Level-C stores one search window (avoiding the retransmission of overlapping search window regions accessed by neighboring MBs in the same line) and Level-D a search window stripe for the whole frame. Observe that Level-D requires a extremely large on-chip memory for large search window or frame size. As Level-C presents a reasonable trade-off between external memory access and on-chip memory size it was extended in Level-C+. Level-C only exploits the data reuse between horizontal neighboring MBs. Level-C+ proposes to increase the vertical on-chip storage to include the search window of the MB line below. This allows exploiting the vertical data reuse at the cost of increased on-chip memory and out-of-order processing (two MB lines are processed using double-Z order).

In Wang et al. (2009) a bandwidth-efficient H.264 ME architecture using binary search is proposed. This solution employs a frame-level preprocessing that downsamples the image twice in a factor 2. It results in three images (or three layers), the original image, the downsampled image, and the twice downsampled image. After that, a search is performed in the three layers. This technique is also modified to allow parallel processing and easy hardware implementation. A hardware architecture is presented targeting low power through low memory access, efficient hardware utilization, and low operation frequency.

A complete MVC encoder targeting low-power operation is presented in Ding et al. (2010) employing eight pipeline stages, dual CABAC, and parallel MB interleaving. A cache-based solution is used for the search window reading along with a specific prefetching technique. The cache tags are formed by the frame index and x and y block position. Also, each cache entry stores an image block (instead of words like in generic caches) following the same concept proposed in Zatt et al. (2007). The search is constrained to a [±16,±16] search window with a predicted center point. The ME/DE architecture is described in more details in the previous work

from the same group (Tsung et al. 2009). This approach might lead to quality loss when the center point prediction is not accurate. Also, the authors ignored the fact that fast ME/DE schemes already consider this information to start the search. The MVC encoder is able to real-time encode four views HD720p at 317 mW.

Generally, the search window-based data reuse approaches suffer from excessive leakage resulting from big on-chip SRAM memories required 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 (Xu and He 2008). In this case, even considering asymmetric search windows incurs in large on-chip storage overhead, thus suffering from significant leakage.

The authors in Shim and Kyung (2009) 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 search window centered prediction is employed for data prefetching while the size of search window is dynamically adjusted at frame level using the size of motion vectors found in previous frames. The data reuse scheme Level-C is employed.

A data-adaptive structured search window scheme is presented in Saponara and Fanucci (2004). An adaptive window size approach is proposed considering the spatial/temporal correlation of the motion field. If the vectors of the current and past frames do not exceed a given value, there is no need to search in a region larger than this vector size and the fetching of a reduced window is necessary. In case the window is too small and the error starts do increase, a test detects it and the search window is increased regardless of the neighborhood. This solution leads to reduced external memory access, but its potential for on-chip memory reduction is not discussed.

The work in Chen et al. (2007) proposed a candidate-level data reuse scheme and a Four-Stage Search algorithm for ME. Firstly, multiple search start points are predicted from the neighboring MBs motion activity. The predicted points are evaluated and the best one is selected for a Full Search around its position. A ladder-shaped data arrangement is also proposed in order to support random access for the proposed algorithm. The candidates parallel processing is performed using a systolic array.

In Tsai et al. (2007) 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 (Tsai et al. 2007), however, is limited to a fixed Four Step Search pattern and it does not consider disparity estimation and power gating.

2.7 Energy/Power Consumption Background

Before moving to the discussion related to energy-efficient algorithms and architectures it is necessary to understand the sources of energy consumption and how they might be reduced. Moreover, the energy consumption is directly related to the hardware implementation and only indirectly related to the algorithms. However, it is possible to design algorithms able to result in energy reduction at the hardware level by reducing computational effort, processing time, memory access, etc.

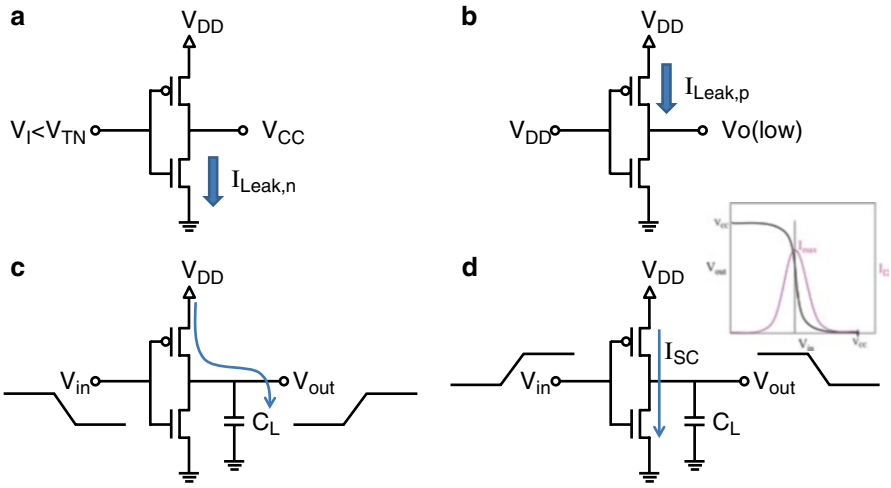

Fig. 2.17 Energy/power dissipation sources

In Fig. 2.17 are represented the three main power dissipation sources for CMOS circuits using an inverter as example: leakage current (static), switching power (dynamic), and short circuit current (dynamic). Eq. (2.5) shows the total power in terms of these three components. The static power dissipation is a result of the leakage currents. Consider Fig. 2.17a where the input voltage (V_I) is lower than the NMOS transistor threshold voltage (V_{TN}). In this case, an ideal inverter NMOS transistor does not conduce any current. However, real MOSFET transistors cannot completely block this current, the so-called leakage current. The closer V_I is to V_{TN}, the stronger the leakage. The same happens to PMOS transistors when a $V_I > V_{TP}$ is applied to the gate (Fig. 2.17b). The leakage power for the case represented in Fig. 2.17b is calculated by Eq. (2.6).

The dynamic power is composed of two components: the switching power (Fig. 2.17c) and the short circuit power (Fig. 2.17d). Equation (2.7) defines the switching power that linearly depends on the load capacitance (C_L, that depends on the fanout of the device), the source voltage V_{DD}, the frequency of operation (f), and the frequency of switching of that device (α). It represents the energy that is charged in the load capacitance and later drained to the ground. Note that only after two switches the energy is actually drained; in the first time instant (shown in Fig. 2.17c) the capacitance CL is charged and in the second time instant (after another switch) the energy is drained from C_L to ground. It justifies the ½ factor in Eq. (2.7). The short circuit current happens while the input signal changes V_{DD}-GND or GND-V_{DD}. There is a given input voltage where both PMOS and NMOS transistors are conducing and a current is drained directly from V_{DD} to the ground. It is depicted by the current in Fig. 2.17d and the short circuit power is defined by Eq. (2.8). The total energy drained is the total power along the time (t) as represented in Eq. (2.9). Other power dissipation sources (such as gate leakage) exist in the CMOS devices, but they are omitted in this short overview for simplicity reasons.

As can be seen from this overview it is possible to reduce both static and dynamic power. For instance, reducing the computation reduces the dynamic power once α is reduced. If frequency scaling is used, f is also reduced. Moreover, if voltage scaling is used the dynamic energy is reduced in a quadratic order because V_{DD} is reduced. For leakage reduction, circuits featuring multiple thresholds are used. Hardware support is required; however, application knowledge and energy-aware control algorithms are required to accurately control thresholds, frequency, and voltage:

$$P_{Total} = P_{Leak} + P_{Switch} + P_{Short}, \tag{2.5}$$

$$P_{Leak} = I_{Leak} \times V_{DD}, \tag{2.6}$$

$$P_{Switch} = \frac{1}{2}\lambda \times f \times C_L \times V_{DD}^2, \tag{2.7}$$

$$P_{Short} = I_{Short} \times V_{DD}, \tag{2.8}$$

$$E_{Total} = P_{Total} \times t. \tag{2.9}$$

2.8 Energy-Efficient Algorithms for Multiview Video Coding

2.8.1 Energy-Efficient Mode Decision

The mode decision is one of the main contributors for the MVC high complexity and consequent energy consumption. The optimal solution using the exhaustive RDO-MD requires the evaluation of all possible inter-prediction and intra-prediction modes defined by the standard. Such solution is not feasible for real-world implementations. Thus, there is a need to reduce the number of evaluated modes during the coding process. Statically defining modes to be tested does not perform well due to changing coding parameters and video input characteristics. For this reason, it is necessary to dynamically define the most probable coding modes using the run-time available data. Figure 2.18 shows a hypothetical fast MD scheme which selects a few candidate modes out of all possible modes. Current solutions, as detailed along this section, use information extracted from the video content (texture, edges, brightness), coding mode history, video geometry, etc.

Several fast MD schemes have been proposed to reduce single-view H.264 complexity, such as fast I-MB MD, fast SKIP MD, fast P-MB MD, and the combination of the above. These fast mode decisions of H.264 may be deployed for MVC. However, they will perform inefficiently for the non-anchor frames as they do not exploit the inter-view correlation and the knowledge of GDV.

Recently, multiple fast MD schemes have been proposed for MVC (Peng et al. 2008a; Lee et al. 2008; Han and Lee 2008; Shen et al. 2009a, b, 2010a; Ding et al. 2008a; Zeng et al. 2011; Chan and Tang 2012) considering the GDV, camera geometrical properties, inter-view correlation, and early SKIP prediction.

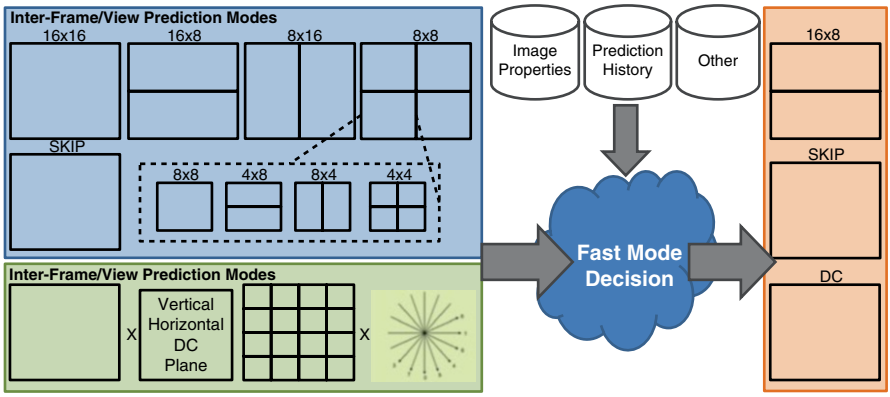

Fig. 2.18 Fast mode decision example

The authors in Lee et al. (2008) proposed an object-based mode decision that uses image segmentation to evaluate different prediction modes for foreground and background regions. The image is segmented using a motion-based approach, considering the vectors size and the SAD in relation to the collocated block (in the same relative position). In case the motion vector significantly defers from the vector average (respecting a threshold) and the SAD exceeds a given value the region is considered as a foreground object; otherwise it is a background. A region growing is used to merge the foreground objects. The foreground regions are coded using DE, while the background are coded using ME. The boundary MBs are coded using the exhaustive RDO-MD.

A fast mode decision based on GDV is presented in Han and Lee (2008). In this scheme the base view—encoded using exhaustive RDO-MD—is used to segment other views in foreground and background regions. The coding modes of the base view are used to classify the image regions. SKIP and Inter 16×16 MBs are defined as background while the remaining modes are considered foreground objects. As the objects present displacement between views, the GDV is used to displace the classified regions as well. Finally, the foreground regions are encoded using exhaustive RDO-MD and the background regions are encoded using big block sizes.

The fast mode decision scheme of Shen et al. (2009a, b) considers the information of reference view to classify the current MB in three complexity classes. For that, the authors propose a mode complexity metric (MDC) defined as the sum of each mode complexity in a 3×3 MBs window. SKIP and Inter 16×16 have "0" complexity, Inter 16×8 and 8×16 have "1" complexity, and Inter 8×8 (or smaller) and Intra have "2" and "3" complexity values, respectively. If the MDC is smaller than a given threshold (T0), that regions is classified as *simple*. In the opposite, if MDC exceeds another threshold ($T1>T0$) it is classified as *complex*. Regions presenting MDC between these thresholds are defined as *medium* complexity. The *simple* regions test only Inter 16×16 mode. *Medium* regions evaluate Inter 16×16, 16×8, and 8×16 modes. *Complex* MBs are encoded using the exhaustive RDO-MD.

In Zeng et al. (2011) a fast mode decision approach is proposed based on the classification of the current MB according to its motion activity based on the coding

modes of the base view. Firstly, the five motion-activity classes are defined in relation to the coding modes. SKIP belong to the motionless class (1). Slow motion class (2) is defined for SKIP and ME 16×16. ME 16×8 and 8×16 are considered Moderate Motion (3). Fast motion regions (4) are defined by ME 8×8 or smaller. Finally, DE and Intra define High texture with fast motion or scene cuts (5). The mode correlation-based mode decision (MCMD) metric is defined and calculated using the 3×3 collocated MB window. Within this 3×3 neighboring MBs, each neighbor MB has an offline defined weight. This MCMD metric is used to classify the current MB motion activity in one of the classes described above. Independent of the motion activity, the SKIP mode is firstly evaluated and an early termination test is employed. If the SKIP prediction was not effective other modes are evaluated according to the motion class. The same classification described above is used here. For instance, A slow motion MB evaluates only the ME 16×16.

The work proposed in Chan and Tang (2012) exploits the statistical behavior of the RDCost for the different coding modes along with the motion vectors difference in order to speed up the MVC encoding. In this solution, an interactive mode decision is employed. Based on statistical knowledge showing the ME is used more frequently than DE, the first interaction evaluates only the ME modes (all sizes). If the ME-based prediction is not satisfactory, a second interaction is used to evaluate the ME modes. However, only the block sizes that presented better coding performance for ME are evaluated for DE in the second interaction.

State-of-the-art schemes mainly achieve the complexity reduction via fast MD. However, they do not exploit the full space of neighborhood correlation in all spatial, temporal, and view domains. These schemes deploy fixed thresholding (Han and Lee 2008; Shen et al. 2009a, b) and, consequently, are unable to react to the changing QPs (i.e., changing bitrates). Moreover, in their worst case, state-of-the-art schemes—like (Han and Lee 2008; Shen et al. 2009a, b)—check all prediction modes, thus falling back to the exhaustive RDO-MD. As a result, these schemes provide limited complexity reduction.

In general, state-of-the-art schemes consider reference view encoded using the exhaustive RDO-MD and employ their fast MD scheme on the other views. These schemes prioritize the frames from the base view and the encoded quality of other views relies on the one encoded using exhaustive RDO-MD. This might lead to meaningful prediction error increase for the last views.

2.8.2 Energy-Efficient Motion and Disparity Estimation

To find a single optimal/good matching block the ME/DE performs several block-matching operations in multiple reference frames. Additionally, this search is replicated for multiple block sizes defined by the MVC standard. However, there are search directions (ME or DE), reference frames, and reference regions that are highly unlikely to provide a good matching. Also, there are suboptimal points that provide similar results at the cost of much reduced searching effort. See the example in Fig. 2.19a. A good matching for the diamond object is available in just one of the

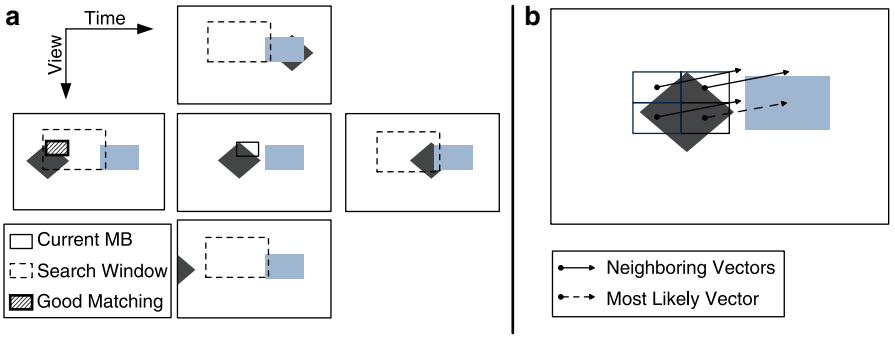

Fig. 2.19 ME/DE search conceptual example

four reference frames, the past temporal reference. In the future temporal reference the diamond is partially occluded by a second object (rectangle). In the disparity references the diamond is either occluded or out of the captured scene. In this scenario there is no need to perform searches in all reference frames, resulting in complexity/energy reduction. Other example is depicted in Fig. 2.19b. Note that the previously encoded neighboring MBs share a similar motion/disparity vector since they belong to the same object. Therefore, the current MB, which also belongs to the same object, is very likely to share a similar vector. This knowledge may be used to reduce the number of search operations by reducing the number of candidate blocks. A wide range of techniques to reduce the ME/DE complexity are available as presented in the following.

State-of-the-art fast ME/DE algorithms employ variable search range based on disparity maps (Xu and He 2008) taking into account the distinct behavior between ME and DE. The work presents an study on how the search window size impacts in the coding efficiency showing the importance of big search windows. However, disparity maps show that it is possible to reduce the effective search window by monitoring the disparity maps. From the disparity maps two parameters named vertical and horizontal scales (VS, HS) are defined. From the parameters the search window is reduced or increased in an asymmetric way, i.e., the search window may assume rectangular shapes. The increase and reduction are done in a factor 2.

In Kim et al. (2007) two strategies are used to predict motion and disparity vectors. One vector predictor used the traditional spatial median predictor from upper, left, and upper right neighboring MBs. The other predictor used the camera geometry and vectors from previously encoded frames to estimate the current vectors. The difference between the two predicted vectors is used to calculate the search window size. A small difference means accurate predictors and a small search window is required. Otherwise, for big differences a larger window is needed.

A fast direction prediction (ME or DE) based on the blocks motion intensity is proposed in Lin and Tang Angela (2009). It exploits the inter-view correlation to predict a search direction for reducing the ME/DE complexity. The base view is encoded using the ME and the frame regions are classified as slow motion if the SAD is smaller than a threshold. Similarly, the anchor frames of all views are

classified according to this strategy. For each MB to be encoded, the collocated MBs from base view and anchor frame are tested. In case both are slow motion, the current MB probably is also a slow motion MB and will be encoded using ME only. If only the base view collocated MB is not slow motion, DE is employed. Other cases require ME and DE processing.

The schemes in Han and Lee (2008), Ding et al. (2008a, b), and Deng et al. (2009) exploit the information from the base view and classify MBs into foreground and background regions. In Ding et al. (2008a, b) a fast ME based on complete DE is proposed. The DE is used to locate the correlated MB in the base view. After that, the coding information extracted from the base view is used to predict the motion vectors and partition sizes for the current MB.

The view-temporal correlation is exploited in Deng et al. (2009) by using the motion information of the first encoded view in order to reduce the complexity of further views. Additionally, disparity vectors from anchor frames are also taken into consideration. Using the geometric relation between the vector from base view and anchor frames, the authors predict the motion and disparity vectors that are used as search start point. A 2×2 refinement is applied around the predicted point. This process is repeated for each search direction.

The inter-view correlation is also evaluated in Shen et al. (2009a, b, 2010a, b) to reduce ME/DE search window. The so-called motion homogeneity is calculated using the collocated motion field from previous frames. If the MB presents a complex motion (homogeneity higher than a threshold) the complete search window is used for searching. Homogeneous motion MBs use a search window reduced in a factor of 4, 1/4 of vertical size and 1/4 of horizontal size. For the intermediate case, the search window is reduced in a factor of 2. Simultaneously, this solution employs a search direction selection. Homogeneous regions employ only ME search while complex motion regions employ both ME and DE. Moderate motion regions use the RDCost information to enable DE search.

Algorithm and architecture for disparity estimation with minicensus-adaptive support is proposed in Chang et al. (2010). A minicensus transform is applied over a pair of frames in two neighboring views to define a matching cost at pixel level. Weights are additionally generated using color distance. The cost and weights are aggregated to find the best disparity between the pair of frames. According to the authors, the two-pass strategy reduces the complexity if compared to a direct approach. The architecture proposed is discussed in Sect. 2.6.

The main drawback of these fast ME/DE algorithms resides in the fact that they do not exploit the full potential of the 3D-neighborhood correlation available in spatial, temporal, and disparity domains. Moreover, even the more sophisticated techniques are dependent on the complete first view encoding. However, it does not scale well for a large number of views as the prediction quality degrades in a hierarchical prediction structure. By encoding one view, the motion field information can be extracted but not the disparity field information (as no inter-view prediction is performed in this case). Therefore, it potentially limits the speedup of disparity estimation. Additionally, most of the techniques use fixed thresholding and thus perform inefficient under varying quantization parameters (QPs).

2.9 Video Quality on Energy-Efficient Multiview Video Coding

Techniques to reduce the complexity and energy consumption of the video encoder (such as fast mode decision and motion/disparity estimation) typically lead to video quality losses. To control the quality losses rate control methods may be employed through QP adaptation. Several rate control schemes are found in the current literature. Mostly they are developed targeting single-view encoders such as H.264. Recently, a few works specific to the MVC standard have been proposed focusing on frame- and BU-level RC. In this section we present an overview of the state of the art on rate control.

In the single-view domain the majority of recent proposals are extensions of the RC implemented in the H.264 reference software that employs a quadratic model for mean absolute differences (MAD) distortion prediction (Li et al. 2003). However, the quadratic model leads to limited control performance, as discussed in Tian et al. (2010). Aware of this limitation, the authors in Jiang et al. (2004) and Merritt and Vanam (2007) propose improved MAD prediction techniques. The scheme presented in Kwon et al. (2007) implements both distortion and rate prediction models while in Ma et al. (2005) the RC exploits rate–distortion optimization models. An RC based on a PID (proportional–integral–derivative control) feedback controller is presented in Zhou et al. (2011). In Wu and Su (2009), an RC scheme for encoding H.264 traffic surveillance videos using regions of interest (RoI) to highlight regions that contain significant information is proposed. In Agrafiotis et al. (2006), RoI is used to highlight preset regions of interests using priority levels. However, single-view approaches do not fully consider the correlation available in the spatial, temporal, and view domains and, consequently, cannot efficiently predict the bit allocation or distortion resulting in inefficient RC performance.

The early RC proposals targeting the MVC encoder are based on simple extension of single-view approaches (Li et al. 2003) and are still unable to fully exploit multiview properties. Novel solutions, however, have been proposed and most of them are limited to frame-level actuation. The solution in Yan et al. (2009b) uses an improved MAD prediction that differentiates the frame types. Intra frames, P and B frames with only temporal prediction, P and B frames with only disparity prediction, and B frames with both temporal and disparity prediction feature distinct MAD prediction equations. Once the MAD is predicted, the target bitrate is predicted for the GGOP, refined to the GOP, and finally defined for each frame. An appropriate QP for each frame is defined based on the target bitrate. This work is extended in Yan et al. (2009a) by employing a technique to define the first QP in the GGOP; it is used to encode the I frame. But these solutions are unable to properly handle the complex HBP structure of MVC limiting the number of input samples and the rate control learning.

The authors of Xu et al. (2011) define an pyramid-based priority structure extracted from the MVC HBP. The higher pyramid levels are used as reference to encode lower pyramid level, e.g., I and P frames belong to the highest level, B frames that refer to I and P frames belong to the second highest level, and so on. The higher levels are prioritized and are encoded using lower QPs (high quality) in

order to reduce error propagation. This solution, however, considers a fixed HBP structure and does not exploit the inter-GOP correlation.

To deal with distinct image regions within a frame there is a need for a BU-level RC. Moreover, in order to find a global optimal solution, a joint frame- and BU-level rate control scheme must be designed. Recent works have proposed solutions for the BU-level RC in MVC. In Park and Sim (2009) is presented a solution that deals with the frame-level and Macroblock (or BU)-level rate control. Firstly, the rate for each view is calculated based on weight parameters defined by the user. After that, the QP for each GOP is defined using the traditional H.264-based approach (Li et al. 2003) followed by a QP refinement for each frame. The frame-level QP definition considers the HBP coding structure to prioritize frames in higher hierarchical levels. A MAD-based strategy is used to calculate the target bitrate at MB level and a rate–distortion model (not described in the paper) is employed to define the QP for each MB.

The authors in Lee and Lai (2011) consider the HVS properties to propose a BU-level rate control solution that prioritizes the regions that are visually more important to the observer. For that, they define regions of interest using the Just-noticeable difference (Liu et al. 2010) metric along with luminance difference and edge information. Depending on the relation between these metrics, the QP is increased or decreased in relation to the initial QP (maximum QP in the neighborhood). However, this solution does not employ feedback-based control and just considers the coding information from one reference frame.

Generally, the available rate control techniques cannot fully exploit the correlation potential available in the spatial, temporal, and view domains of MVC. In addition, they are unable to adapt to multiple HBP structure and cannot employ the inter-GOP periodic behavior for RC optimization. Moreover, at the best of our knowledge, no work has proposed a Rate Control scheme for MVC able to jointly consider frame and BU level in a hierarchical and integrated fashion.

2.9.1 Control Techniques Background

In this section are presented the background concepts required to understand the rate control solution proposed in this monograph. Firstly, are presented the control theory basics behind the model predictive control (MPC) used for the frame-level RC. On the following, we present the statistical foundation supporting the Markov decision process (MDP) that is implemented in our BU-level RC. Finally, the concepts related to reinforcement learning (RL) are introduced.

2.9.1.1 Model Predictive Control

The control theory is a subfield of mathematics originated in engineering to deal with influences in the behavior of dynamic systems (Tatjewski 2010). Several control methods have been proposed ranging from very general to application-specific solutions to cope with a wide range of applications. Control problems specifications

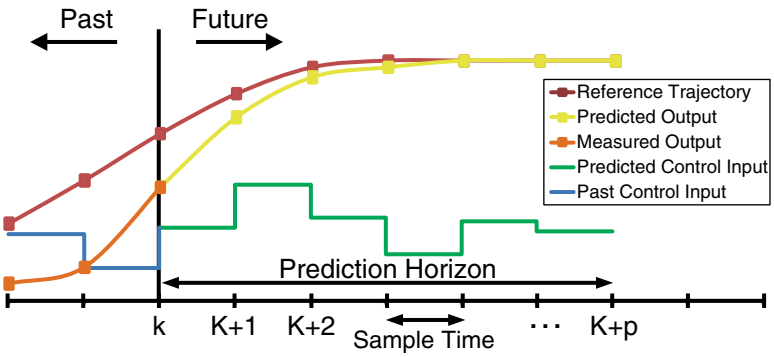

Fig. 2.20 Model predictive control (MPC) conceptual behavior

may significantly vary and the selected control method must ensure the stability of the given system. Thus, the selection of a control method for a given dynamic system may be very challenging. In case the controller does not fit the system it may compromise the stability of the entire system.

Among state-of-the-art control methods, the MPC has gained prominence by being able to accurately predict and actuate on a dynamic multivariable systems. It represents not a single control algorithm but a controller design scheme applicable to distinct systems including continuous or discrete in time, linear or nonlinear, integrated or distributed systems. MPC outperforms conventional feedback controllers (like PID) by keeping explicit integration of input and state constraints while considering state space constrains. Also, MPC can dynamically adapt to new contexts by employing rolling input and output horizons (see more details below).

The main goal of the MPC is to define the optimal sequence of actions to lead the system to a desired and safe state by considering the system feedback to previous states and previously taken actions (see conceptual MPC behavior in Fig. 2.20). To define this sequence of actions the MPC minimizes the performance function presented in Eq. (2.10). It minimizes the cost by defining a set of outputs y based upon a set of inputs u. Where $u[k+i-1|k]$, $i=\{1, \ldots, m\}$ denotes the set of process inputs with respect to which the optimization is performed; u is known as the control horizon or input horizon in the MPC theory. Analogously, $y[k+1|k]$, $i=\{1, \ldots, p\}$ is the set of outputs, named prediction horizon or output horizon (see Fig. 2.20). The control horizon determines the number of actions to find. The prediction horizon determines how far the behavior of the system is predicted. m and p are the size of control/input and prediction/output horizons, respectively. m is the number of measured outputs (history size) used for the optimization process ,while p defines how many outputs are predicted; that is, how many future actions are considered in the optimization processes. k is the horizons index and represents the kth input/output horizon. y^{SP} defines the output set point that limits the prediction horizon:

$$\min_{u[k|k]\ldots u[k+p-1|k]} \sum_{i=1}^{p} w_i \left(y[k+1\,|\,k] - y^{sp}\right)^2 + \sum_{i=1}^{m} r_i \Delta u[k+i-1\,|\,k]^2. \tag{2.10}$$

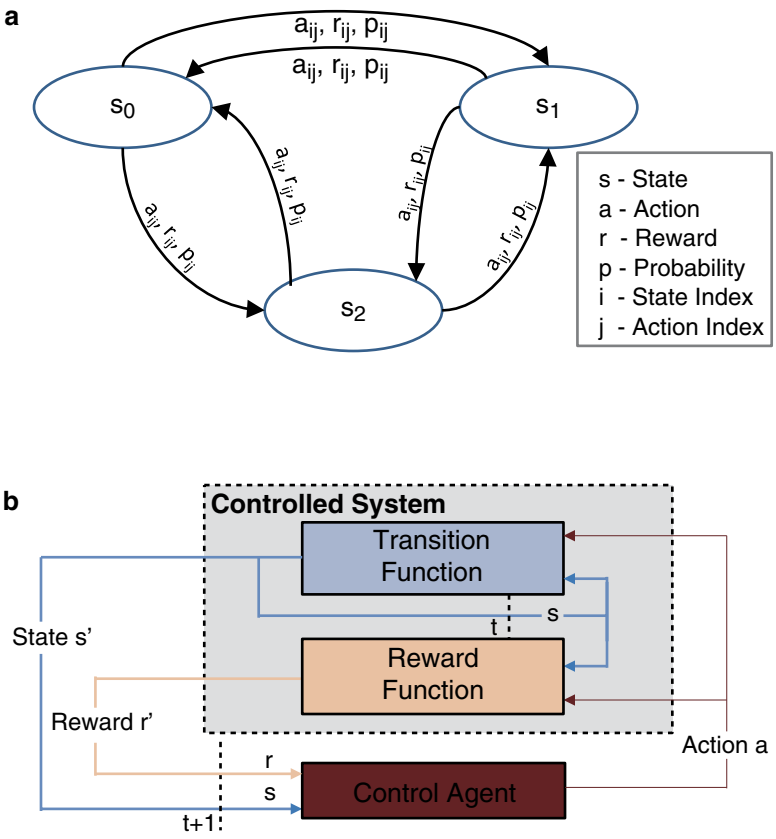

Fig. 2.21 Markov decision process (MDP)

2.9.1.2 Markov Decision Process

The MDP is a mathematical decision-maker framework for systems that outcome partly random and partly controlled by a decision maker (Arapostathis et al. 2003). MDP is a time discrete stochastic control process based on the extension of Markov Chains that adds the concepts of actions and rewards.

The symbolic representation of the MDP is a state machine or an automaton, as depicted in Fig. 2.21a, which evolves in response to the occurrence of events. It is formally defined by 4-tuples (S, A, $P(.,.)$, $R(.,.)$) composed of a finite set of states $S=\{s_0, s_1,\ldots\}$, actions $A=\{a_0, a_1,\ldots\}$, rewards $R=\{r_0, r_1,\ldots\}$, and transition probabilities $P=\{p_0, p_1,\ldots\}$. The S includes all possible states assumed by the controlled system; actions A are the possible acts to be taken by the decision maker in face of a given system state. $P(S)$ is the probability distribution of transitions between system states and, finally, $R(S)$ is the reward related to a given action for a given state. At each discrete time step t the process lays in a state $s \in S$ and the decision maker may choose any action $a \in A$ that will lead the process to a new state $s' \in S$ providing a

shared reward $R_{at}(s, s')$, as shown in Fig. 2.21b. The rewards are used by the decision maker in order to find an action that maximizes, for a given policy, the total accumulated reward, as shown in Eq. (2.11) (where $0 \leq \gamma \geq 1$ denotes the discount factor):

$$\sum_{t=0}^{\infty} \gamma^t R_{at}(s_t, s_{t+1}). \tag{2.11}$$

By definition, the Markov process is considered a controlled Markov process if the transition probabilities $P(S)$ can be affected by an action. Equation (2.12) defines the probability P_a that an action a in the state s at time t will lead to state s' at time $t+1$:

$$P_a = R_{at}(s_{t+1} = s' \mid s = s_t, a_t = a). \tag{2.12}$$

Multiple extensions have been proposed to the MDP in order to best fit to distinct problem classes. For systems where the transitions probabilities or the rewards are unknown a priori, the Reinforcement Learning method may be applied to solve the MDP, as detailed in the following section.

2.9.1.3 Reinforcement Learning

Reinforcement learning model is an agent to improve autonomous systems performance through trial and error by learning from previous experiences instead from specialists, that is, the agent learns from the consequences of actions. In reinforcement learning model the agent is linked to the system to observe its behavior and take actions. RL theory is based on the Law of Effect, that is, if an action leads to a satisfactory state the tendency to produce this action increases. For each discrete time step t the RL agent receives the system state $s \in S$ and rewards $R(S)$ to take an action $a \in A$ that maximizes the reward $R_{at}(s, s')$. This action may lead the system to a new state $s' \in S$ and produce a system output, in terms of a scalar reinforcement value, used to define the new reward $R_{a(t+1)}(s, s')$ according to Eq. (2.13). The general representation of reinforcement learning value given by RL in Eq. (2.14), where U denotes the function that changes the system state from s to s' and h_R denotes the history of reinforcement learning:

$$\mathrm{RL}_{a(t+1)}(s, s') = \mathrm{RL}_{at}(s, s') + \mathrm{RL}, \tag{2.13}$$

$$\mathrm{RL} = U(s, s') + h_R. \tag{2.14}$$

2.9.1.4 Region of Interest

Within a video frame there may exist multiple regions or objects with distinct image properties and distinct importance for the observer. The image regions that are considered, for some reason, more important are called Regions of Interest. In this monograph, we consider all regions of semantically equal importance

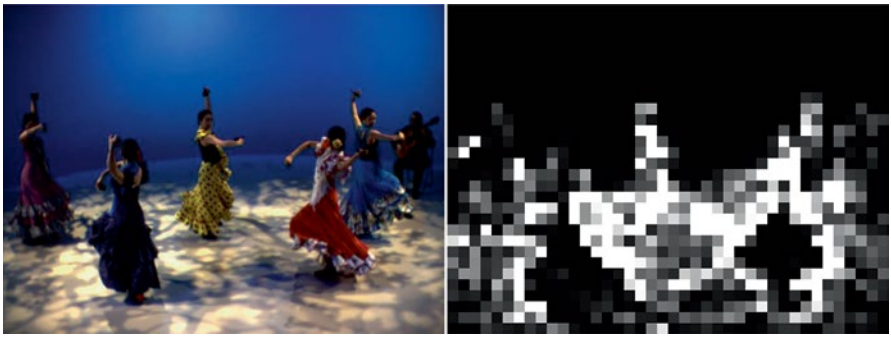

Fig. 2.22 Variance-based region of interest map (*Flamenco2*)

leaving space for application-specific optimizations such as for 3D-surveillance, 3D-telemedicine, etc. However, at the encoding perspective, textured regions tend to have different coding properties at the mode decision and bit allocation perspectives if compared to homogeneous regions. To classify the image regions we use the variance map (Fig. 2.22) to characterize the texture complexity. Variance depicts the degree of dissipation of a given population [see definition in Eq. (4.1)]. In this case, how the pixel values of an image region are distributed. High variance define textured regions (represented by brighter points in Fig. 2.22) while low variance define homogeneous regions (dark regions in Fig. 2.22).

2.10 Summary of Background and Related Works

The MVC is the most efficient video coding standard focusing on 3D-video coding. It is able to provide 20–50 % of coding efficiency increase, if compared to H.264 simulcast, by employing inter-view prediction, the disparity estimation. Mode decision and motion and disparity estimation represent the most complex modules in the MVC encoder and bring big challenges for their real-world implementation.

The implementation of MVC encoders may exploit different multimedia processing architectural solutions. Currently, the most preeminent alternatives are multimedia processors/DSPs, reconfigurable processors, ASICs, and heterogeneous multicore SoCs. Each solution presents positive and negative points. On the one hand, ASICs provide the highest performance and energy efficiency at the cost of no flexibility. On the other hand, multimedia processors/DSPs are totally flexible but deliver low performance and reduced energy efficiency. Heterogeneous multicore and reconfigurable processors provide trade-off points between ASICs and processors. By employing units specialized in each kind of task, the heterogeneous multicore SoCs improve the performance in relation to multimedia processors but typically present issues related to programming and portability. Reconfigurable processors can cover this gap by employing extensible instruction set and defining, at

run time, if regular or custom instructions should be used in that specific time instant. Still, these solutions are unable to meet the performance and energy efficiency required for MVC encoding without application-specific ASIC acceleration. Therefore, considering the current technology, a complete ASIC encoder or heterogeneous SoCs with hardware specific accelerators are seen to be the most feasible solutions for embedded mobile devices.

Multiple proposals targeting on complexity and energy reduction for the MVC are available in the current literature. These contributions are centered in two abstraction levels: the algorithmic and architectural levels. At the coding algorithms perspective, complexity reduction is most frequently addressed at the mode decision and motion and disparity estimation because they represent the most complex MVC blocks. The mode decision solutions used distinct side information in order to reduce the number of coding modes tested during the coding process. Video properties such as texture, edges, luminance, and motion/disparity activity are used to predict the most probable coding modes in each image regions. Additionally, extensive analysis has been done to learn how neighboring views and frames are correlated. This correlation is also useful to predict the coding modes. As the ME/DE spends about 90 % of the total encoding time, the same kind of information is used to predict the most probable motion and disparity vectors and reduce the ME/DE complexity. However, the related works do not fully exploit the correlation available within the 3D-neighborhood and perform badly under content changing scenarios. Moreover, these solutions are not developed considering the energy perspective and cannot react to battery-level changing situations by dynamically adapting the complexity to the available energy.

Generally, the complexity reduction techniques lead to uncontrolled quality degradation and coding efficiency losses. The rate control becomes a key task in order to minimize this complexity reduction drawback. The majority of rate control solutions currently available target the H.264 or are simple extensions from H.264 solutions. The few rate control algorithms designed for MVC focus only on frame-level or basic unit-level actuation levels. Additionally, theses algorithms do not use the intra- and inter-GOP bitrate correlation in the 3D-neighborhood.

At the hardware architectural perspective ME/DE is the most studied MVC coding block. The ME/DE is a processing and memory-intensive task requiring massively parallel processing and efficient memory access and management. The resulting high-energy consumption is mainly related to external memory access and on-chip video memory size. Diverse related works propose ME/DE processing hardware architectures, memory hierarchies, and data reuse techniques. However, they share limitations related to the complexity reduction algorithms implemented (leading to quality losses), excessive external memory accesses, or large on-chip memory resulting in high energy. Moreover, most of the available architectures lack the ability to dynamically adapt its operation according to changing coding parameters or video content characteristics.

Therefore, there is a demand for novel and energy-efficient MVC encoding solutions able to significantly reduce energy consumption under changing video and system scenarios. For this reason, this monograph targets on jointly addressing the energy issues at algorithmic and architecture levels while sustaining the video quality.

http://www.springer.com/978-1-4614-6758-8

3D Video Coding for Embedded Devices
Energy Efficient Algorithms and Architectures
Zatt, B.; Shafique, M.; Bampi, S.; Henkel, J.
2013, XIX, 204 p. 126 illus., 112 illus. in color.,
Hardcover
ISBN: 978-1-4614-6758-8