

# Chapter 1

## Introduction

Digital multimedia systems play a more and more important role in today's digital world. Ubiquitous multimedia has become one of the major goals of current technology development. In this chapter, we will first introduce the trends in image and video coding algorithms, which presents how image coding is evolved from discrete cosine transform (DCT)-based block coding to discrete wavelet transform (DWT)-based bit-plane coding, and how video coding is evolved from close-loop motion compensated prediction (MCP) to open-loop motion-compensated temporal filtering (MCTF). Then general design considerations and methodologies of VLSI implementation for multimedia systems are discussed. Lastly, the operation hierarchy of DWT and MCTF is identified, which is a guideline of the VLSI architecture and memory analysis for DWT and MCTF.

### 1.1 Trends in Image Coding Algorithm

#### 1.1.1 *DCT-based block coding*

Before this century, image coding algorithms were mainly based on DCT block coding as shown in Fig. 1.1(a), such as the popular JPEG standard [1]. Usually, the image is separated into  $8 \times 8$  blocks, and DCT is performed on them. The two-dimensional (2-D) DCT can analyze 2-D image signals into nearly uncorrelated 2-D frequencies to reduce source redundancy. The performance of DCT is very close to the optimum Karhunen-Loeve Transform (KLT) but requires less computational complexity [2]. For a higher compression ratio, a lossy image coding is usually applied by quantizing DCT coefficients. The quantization is the only component that results in image quality distortion. However, the distortion can be made visually unnoticed

by considering human visual perceptibility. For example, people are more sensitive to low frequency distortion than high frequency distortion, such that the quantization steps for higher frequencies can be enlarged more for a better compression capability. After quantization, the redundancy among frequency coefficients is further reduced by entropy coding. The coding symbols are usually established by zig-zag scanning the quantized frequency coefficients. Then they are encoded by variable length coding (VLC), such as Huffman coding or arithmetic coding.

For increasing the coding performance, the DC frequency coefficients are usually through differential pulse code modulation (DPCM) first before quantization and entropy coding. The DCT-based block coding highly utilizes the correlation and data dependency among DCT coefficients to achieve acceptable coding gain for popular manipulation. However, it is hard to provide many other functionalities, such as spatial or quality scalable coding and error-immune transmission. If some part of bitstream is missed, the image is very likely to be crashed because of DPCM and zig-zag scanned VLC.

### 1.1.2 *DWT-based bit-plane coding*

In the last decade, the DWT-based image coding matured to provide excellent coding gain and many other functionalities. The DWT transform is basically a frame-based computation. The image can be separated into many tiles, on which DWT are performed independently. The tile size is an encoding issue and could be large than  $256 \times 256$  for avoiding blocking artifacts [3]. As shown in Fig. 1.1(b), the tile (or image) is transformed into hierarchically structured DWT subbands. The subband coefficients are encoded in a bitplane-by-bitplane way, instead of the word-by-word way in JPEG. There are two kinds of subband coding schemes that exploit inter- and intra-subband redundancy, respectively. The inter-subband coding is to explore the redundancy among subbands. They usually use a tree structure to utilize the similarity for high coding performance, such as EZW [4] and SPIHT [5]. On the other hand, the intra-subband scheme only utilizes the local redundancy inside each subband by context-adaptive coding, like EBCOT [6]. The EBCOT algorithm is also adopted in the JPEG 2000 standard [7], which can achieve about 2dB peak-signal-to-noise-ratio (PSNR) gain than JPEG [3]. Moreover, one DWT-based coding algorithm that combines both intra- and inter-subband schemes, called EZBC, can achieve about 0.5dB coding gain more than EBCOT [8].

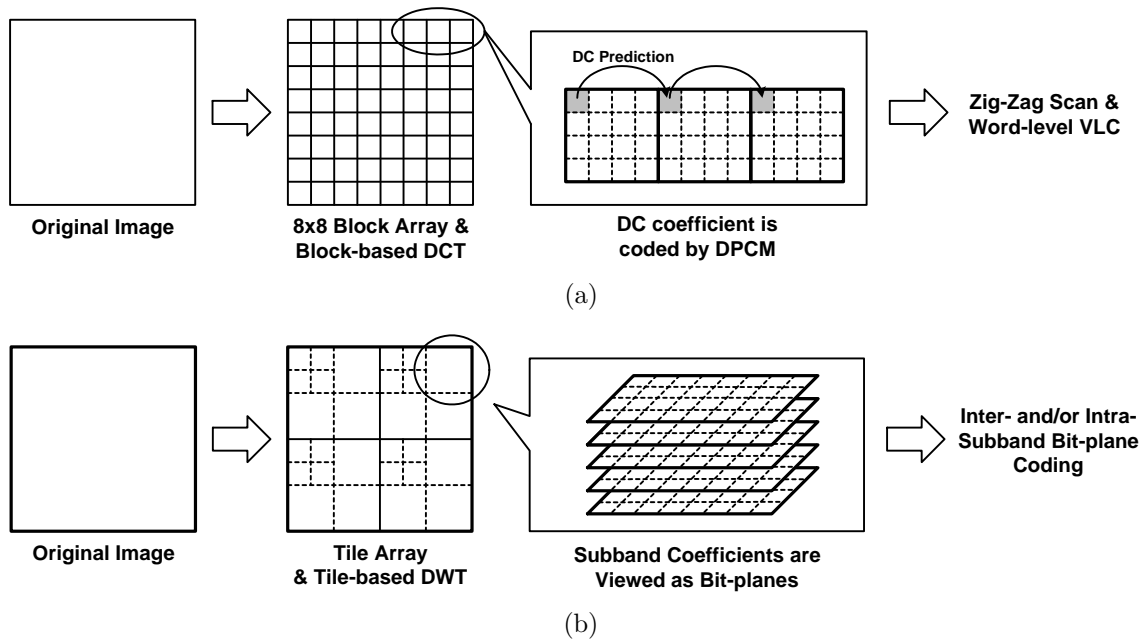


Fig. 1.1 Transform-based image coding flows. (a) DCT-based block coding. An image is separated into small blocks, and DCT is performed on them. For higher coding performance, the DC coefficients cross blocks are usually through DPCM before quantization and word-level entropy coding. (b) DWT-based bit-plane coding. An image is separated into large tiles. After performing DWT on them, the DWT coefficients are encoded by bit-plane-based entropy coding algorithms.

All above-mentioned DWT-based bit-plane coding algorithms can produce embedded bitstreams for quality scalability. The bit-planes can be refined from the most significant bit (MSB) to the least significant bit (LSB) sequentially. They can produce single bitstream when encoding, and the decoder derives more visual quality when receiving more bitstream for more bit-planes. Besides, the intra-subband coding can also provide spatially scalable bitstreams that utilize the time-frequency decomposition property of DWT. For example, the decoder can derive the one-fourth size image by only receiving the bitstream of the low-pass-low-pass (LL) subband. Furthermore, the independent coding unit of EBCOT, a code-block, can be smaller than a subband, which is usually of size  $32 \times 32$  or  $64 \times 64$ . Thus, one missed code-block will not hurt the decoding of other independent code-blocks, which makes a robust error-immune transmission. The high quality embedded bitstream of great error immunity is the reason why JPEG 2000 can provide better communication capability than JPEG.

## 1.2 Trends in Video Coding Algorithm

### 1.2.1 *Close-loop motion-compensated prediction*

Existing hybrid video standards, such as MPEG-1/2/4 [9, 10, 11] and the emerging H.264/AVC [12], mainly consist of a close-loop motion-compensated prediction (MCP) scheme and a transform-based texture coder. The MCP is used to reduce temporal redundancy of video frames, and the texture transform is adopted to reduce spatial redundancy. The “close-loop” means it uses the reconstructed frames to predict the current frame, which forms a feedback loop as shown in Fig. 1.2(a). The “MCP” means image signals of the current frame are compensated by those in the previously reconstructed frames with proper motion models. The block-based motion model is usually used because of the rigid object motion assumption. The close-loop MCP scheme has been highly optimized for the compression efficiency in the last decade, and the H.264/AVC is a landmark of this development. However, for many video applications in the present and the future, the spatial, temporal, and quality scalabilities are more in demand.

The scalability means we can have multiple adaptations for one video bitstream, such as different frame sizes, frame rates, and visual qualities. The close-loop MCP scheme is hard to provide scalabilities while maintaining a high compression efficiency because of the drift problem, like

MPEG-4 FGS. The drift occurs when the encoder and decoder have different reconstructed frames, which can create serious error propagation in the close-loop MCP scheme. But the encoder and decoder inevitably have different video sequences, when the scalability is provided. For overcoming the drift problem, the compression efficiency will be greatly degraded and become unacceptable when there are many scalability layers. This is due to using the base layer that has worse visual quality as reference frames and adopting DCT coefficients for scalable quality coding.

### 1.2.2 *Open-loop motion-compensated temporal filtering*

The open-loop interframe wavelet coding scheme becomes a good alternative for scalable video coding. The concept is to perform wavelet transform in the temporal direction. But the coding performance is unacceptable without motion compensation (MC). In 1993, Ohm introduced a block-based displacement interframe scheme using the Haar filter [13]. However, the compression efficiency is still not comparable to existing MCP video standards until the lifting-based motion-compensated temporal filtering (MCTF) is proposed and the longer tap wavelet filters, like 5/3 filter, are used [14, 15]. The details of the development of MCTF can be found in [16]. The MCTF is a breakthrough of video coding algorithms, which breaks the close loop for efficient scalable coding as shown in Fig. 1.2(b). It uses original frames or filtered frames, instead of reconstructed or coded frames, as reference frames for MC. Any drifted mismatch will only propagate locally.

MPEG has identified a set of applications that require scalable and reliable video coding technologies. After evaluating the response to call for proposals on Scalable Video Coding (SVC) [17], it has been shown that there is a new and innovative video technology that MPEG can bring to industry in a future video standard [18]. The scalable extension of H.264/AVC with MCTF has been adopted in the Working Draft (WD) 1.0 of SVC [19]. It is a hybrid open-loop and close-loop video coding scheme. The lifting-based MCTF is the core technology to provide scalable video coding. The MCTF can provide a variety of efficient scalabilities because the drift problem of traditional close-loop prediction scheme is prevented by the open-loop structure. It also can further increase the compression efficiency of H.264/AVC [20, 21].

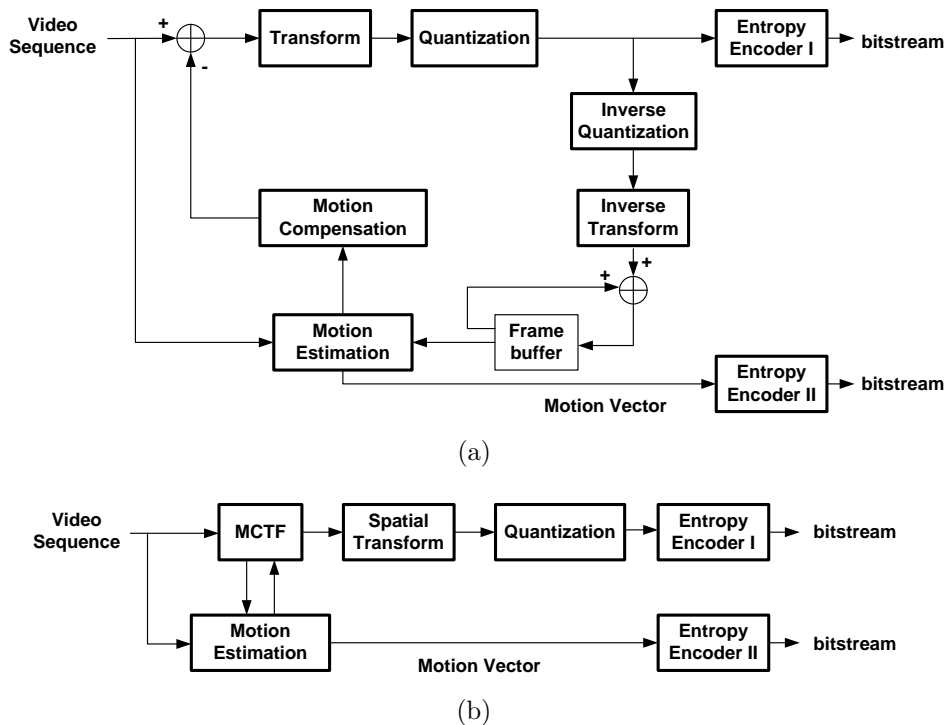


Fig. 1.2 Hybrid video coding flows. (a) Close-loop MCP scheme. The MCP forms a close-loop that also includes forward/inverse texture transform and forward/inverse quantization. (b) Open-loop MCTF scheme. The temporal redundancy is reduced by temporal filtering with motion compensation. No feedback loop is required.

### 1.3 VLSI Design Consideration of Multimedia Systems

Low cost and low power hardware with sufficiently high performance is extremely essential for image and video coding applications to be popular. Thus, efficient hardware implementations in VLSI are of vital importance. However, image and video coding algorithms usually require very high computational complexity and data access. In the following, the design methodologies for hardware architectures of image and video coding are introduced.

The optimization of hardware design for image and video coding systems can be achieved by considering two different levels of architecture design: system design and module design. The former decides the whole system architecture and the relationship between modules. The latter is to optimize each module according to the allocated resource and constraints.

#### 1.3.1 *System design consideration*

The system architecture is usually designed by use of computation analysis and data access analysis that consider computing and data issues, respectively.

##### 1.3.1.1 *Computation analysis*

The computation analysis is to classify the coding tools of the adopted coding algorithm into different level computational characteristics and choose the suitable implementation types. It further includes computational characteristic and complexity analysis.

Figure 1.3 shows the computational characteristic analysis, which categorizes computation into three different levels of operations. On the other hand, the computational complexity analysis is to evaluate the complexity of each coding tool by task profiling. The general result is that low-level operations require more complexity. The low-level operation represents highly regular computation and predictable computational flow. It is suitable to be implemented by a dedicated hardware, because its complexity is usually very high and the regular computation can be easily accelerated via parallel processing. The high-level operation represents highly irregular computation and unpredictable computational flow. However, its complexity is usually much lower than the low-level operation. Thus, it is suitable to be implemented using programmable design. Between these two ex-

tremes, the medium-level operation is preferred to be implemented by use of configurable architecture that can be on-the-fly adapted according to data-dependent decisions.

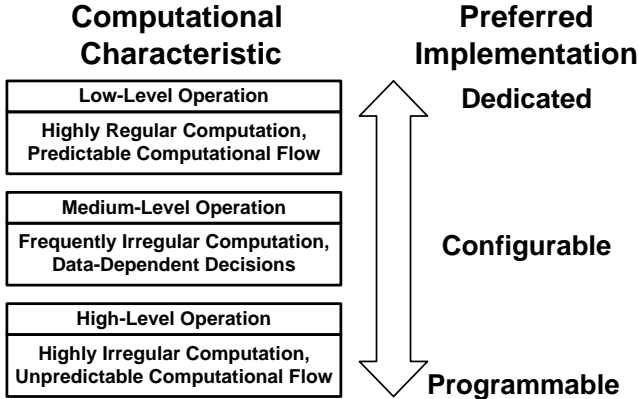


Fig. 1.3 Computational characteristic analysis and the corresponding preferred implementation.

### 1.3.1.2 Data access analysis

The data access analysis is used to decide how data are transferred in the system. It analyzes how data should be stored for access (memory management) and how data are transferred between modules (bus interconnect). The storage and access issue is to provide a good memory management. As described in [22], memory bandwidth and on-chip memory capacity are limited factors for many multimedia applications, and today on-chip memory already occupies more than 50% of total chip area in many applications. Good memory management becomes necessary for successful system-on-a-chip solutions.

The memory management is to provide an efficient and balanced memory hierarchy that consists of off-chip memory, on-chip memory, and registers [23], as shown in Fig. 1.4. Different memory types have different features. Off-chip memory, usually DRAM, offers a large amount of storage size but consumes the most power. The off-chip memory and I/O access may dominate the power budget. To solve this problem, many techniques have been developed. For example, embedded DRAM [22] is developed to reduce the I/O access by integrating large on-chip DRAM. However, the

embedded DRAM technology is not very mature because the yield issue and many physical design challenges are still needed to be solved. Even integrating large embedded DRAM, the access power is still larger than smaller on-chip SRAM. Besides, embedded compression (EC) is usually adopted to reduce the off-chip memory bandwidth and size for video decoders [24, 25, 26], and few EC algorithms are designed for video encoders [27]. EC is to compress the transmitted data on-the-fly to reduce the data amount, but video quality could be degraded if lossy EC is applied.

The on-chip memory, usually implemented by SRAM, can provide faster access and less power-consumption than off-chip memory, but the memory cell size is larger. It also occupies much on-chip die area for implementing multimedia coding systems. Registers can be faster than on-chip memory and provide the most flexible data storage.

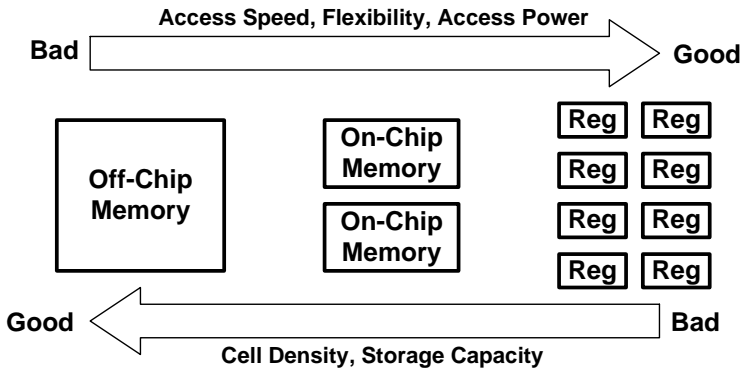


Fig. 1.4 Memory hierarchy consists of off-chip memory, on-chip memory, and registers.

Memory management can be organized from two different levels: algorithm-level and architecture-level. The algorithm-level memory hierarchy optimization is to modify the coding system algorithm to improve some system parameters, like power or area, but some other parameters, like coding performance, become the trade-off. For example, EC belongs to one algorithm-level memory improvement, in which degraded visual quality could be traded by reducing off-chip memory bandwidth. On the other hand, the architecture-level memory organization is to optimize the memory hierarchy from modifying hardware architecture. For example, on-chip memory is usually used as a cache to replace regular off-chip memory access for reducing off-chip memory bandwidth.

The interconnect issue is to decide how to allocate global bus and dedicated connection between modules. The global bus provides flexible configuration and saves the interconnect area. The dedicated interconnect is to provide high throughput and efficient communication between highly related modules.

### 1.3.2 *Module design consideration*

After the system architecture is defined, every module can be designed by use of algorithm-level and architecture-level optimization. The algorithm-level optimization is mainly to optimize the rate-distortion quality under given complexity constraints, or vice versa. For some coding tools, such as motion estimation (ME) and rate control, more computation power usually results in the better visual quality. How to provide a acceptable quality and minimize the required computational complexity is the design challenge.

The architecture-level optimization is to perform data flow smoothing and to balance scheduling and timing control for determined algorithms. Many VLSI implementation techniques [28], such as pipelining, folding, unfolding, and systolic array mapping, can be used for this purpose. Besides computational consideration, good memory management for some critical module, like ME, is also very important. There have many data re-use schemes for different memory management strategies been proposed for ME, such as Level A to D re-use schemes [29, 30].

## 1.4 Book Outline

The chapters in this book present VLSI implementation issues for DWT and MCTF that have led revolutions for image and video coding algorithms, in which architecture design and memory analysis will be discussed. Figure 1.5 shows the operation hierarchy and corresponding design challenge for 1-D DWT, 2-D DWT, and MCTF, which is an outline of this book. Chapters 2 and 6 will give detailed introductions to DWT and MCTF.

The 1-D DWT belongs to pixel-level operation, and it is the basic processing element for 2-D DWT. In Chapter 3, various VLSI architectures of 1-D DWT are introduced and compared. The design effort is focused on area and speed efficiency. The different 1-D DWT design categories will be discussed from different mathematical formulations.

The 2-D DWT becomes frame-level operations. VLSI architecture de-

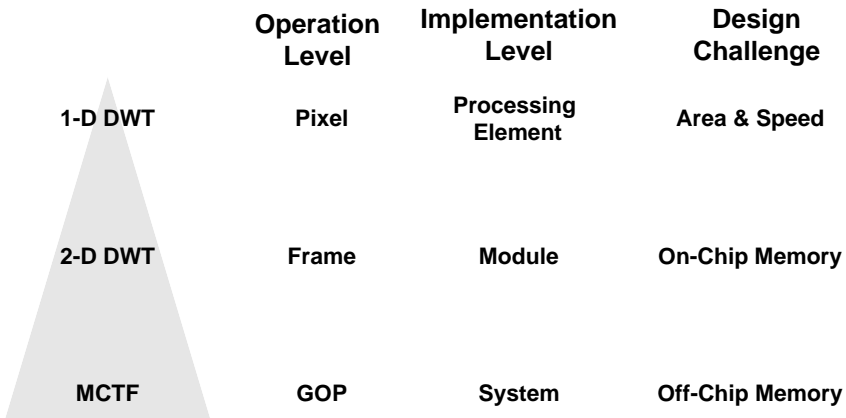


Fig. 1.5 Operation hierarchy and design challenges of 1-D DWT, 2-D DWT, and MCTF.

signs for 2-D DWT are presented in Chapter 4. The trade-off between off-chip memory bandwidth and on-chip memory area is the critical point. When the memory management strategy for 2-D DWT module is decided after considering system requirement, how to optimize the implementation of on-chip memory is the design challenge. We will introduce many memory management schemes for 2-D DWT and discuss on-chip memory implementation issues in detail.

Chapter 5 contains the design examples of 2-D DWT in JPEG 2000 encoding systems. The design issues of 2-D DWT under the constraints of system scheduling will be discussed. Two types of JPEG 2000 encoding scheduling will be shown and the corresponding DWT architecture will be introduced.

Furthermore, the MCTF operates in the group-of-picture (GOP) level, which performs DWT through temporal direction with MC. The ME is the most important computation for finding the best MC blocks in the commonly used block-based motion model of existing video coding standards. In Chapter 7, algorithms, VLSI architectures, and memory management schemes for ME will be reviewed. Chapter 8 considers the design issues for MCTF. Due to the huge amount of data access and computation, the design challenge of MCTF becomes a system-level consideration. The system issues, especially off-chip memory requirement, will be discussed and explored.