ECE462 Multimedia Systems

Laboratory Assignment 4: Video Encoding

Spring 2019

TA: Keerthi Nelaturu

Preparation

1 Instructions

You must read the background material included in this handout *before* you begin your first Lab 4 session.

During the labs you will be required to demonstrate various results to the TA – these will be outlined in the procedure handouts and will be worth a total of 40 marks. It is your responsibility to ensure the TA sees and marks your work.

Note: to execute the lab, you will also need to familiarize yourself with the textbook sections 10.3.1 and 10.3.2.

2 Introduction

Video and audio coding standards define tools and guidelines to represent video and audio data for easy navigation, storage and transmission. Standards such as MPEG-1 and MPEG-2 have enabled many consumer products. For example, MPEG-1, completed in 1992, has been used for network video conferencing, and MPEG-2, introduced in 1994, is the basis for DVDs, digital TV and HDTV.[1] A new generation of standards such as MPEG-4, AVC[2] and VC-1[3] have been developed more recently to address new multimedia applications such as Internet streaming, video scalability, and personalized entertainment systems.

The objective of Lab 4 is to understand the video encoding pipeline and to implement some components of MPEG-4 baseline profile encoder.

[1] Depending on the country, the format of HDTV might be different. Some HDTV is also broadcast in MEPG-4 (Europe) and AVC (Japan).

[2] AVC is also known as Advanced Video Coding, H.264 and MPEG-4 part 10.

[3] VC-1 is used by Microsoft's Windows Media Player.

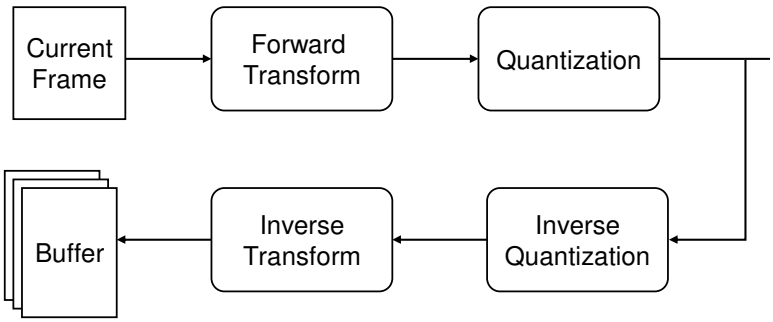

Figure 1: Encoder block diagram for an I-Picture.

3 MPEG-4

The basic structure of MPEG-4 video standard is sometimes referred to as motion-compensated transform coding structure. Each picture or frame is usually partitioned into a number of slices and the slices are coded separately. Each slice consists of a sequence of non-overlapping *macroblocks* (MB). For the luminance channel, an MB is 16×16 pixels in size. For the chrominance channels, an MB can either be 16×16 or 8×8 pixels depending on the sample format. In this lab, we will treat an entire picture as one slice and we will only deal with the luminance component.

A picture can be one of the three types, intra (I), predictive (P) and bi-predictive (B). An I-picture is encoded like a still image, and no motion compensation is used (Fig. 1). P-pictures use the previous I- or P-picture as a reference. Each MB in a P-picture is matched to an area in the reference picture based on the result of a search. Only the prediction difference (the "error" or the residue) and motion vectors are encoded (Fig. 2). B-pictures can use both the previous and the future I- or P-picture as references. The advantage of B-picture is that one can often find a "better" match for a given macroblock than that of using one reference picture. However, this also increases the computation time significantly.

3.1 MPEG-4 Encoder Components

In this lab, you will implement individual function blocks of the MPEG-4 encoder, namely 8×8 forward and inverse transform, quantization and inverse quantization, motion search, and motion compensation.

3.1.1 Forward and Inverse Transform

MPEG-4 utilizes an 8×8 floating point DCT transform which is the same as the one implemented in Lab 2. To encode I-pictures, the transform is applied on the original pixels directly. For P- and B-pictures, the transform is applied on the prediction error.

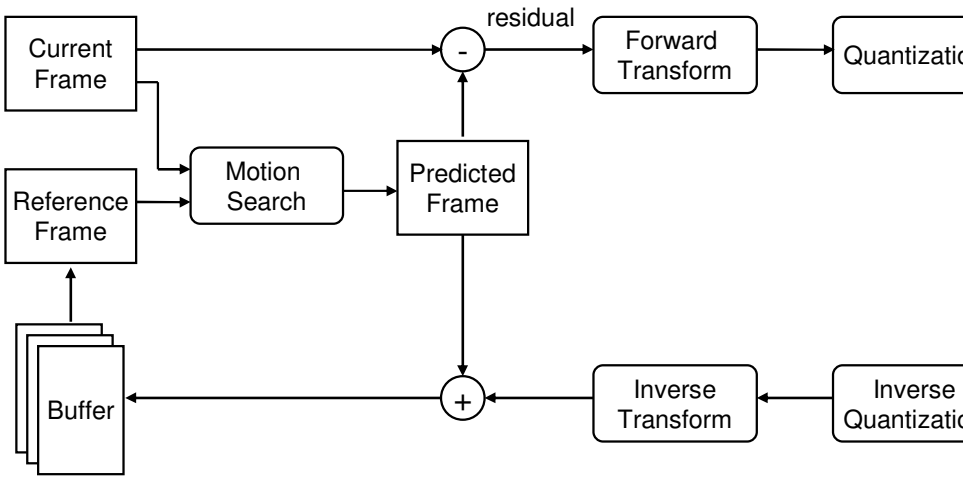

Figure 2: Encoder block diagram for a P-Picture.

The *encoder* needs the inverse transform in order to reconstruct I- and P-pictures to use as reference. This procedure is to ensure that the encoder and the decoder use the same reference pictures to perform motion compensation (otherwise, there would be error propagation at the decoder).

3.1.2 Quantization and Inverse Quantization

Quantization is also referred to as scaling in some of the video standards. MPEG-4 incorporates a quantization matrix plus a quantization parameter (QP). We will use the same mid-tread quantizer used in Lab 2 – the actual quantizer used in MPEG-4 is very similar.

Let $\mathbf{T}$ be the DCT coefficient matrix, the quantized version, $\mathbf{T}^q$, is calculated as

$$\mathbf{T}^q_{[u,v]} = \text{round}\left(\frac{\mathbf{T}_{[u,v]}}{\text{QP} \cdot \mathbf{Q}_{[u,v]}}\right) \tag{1}$$

where $\mathbf{Q}$ is the quantization matrix. The encoder can define a custom quantization matrix $\mathbf{Q}$. MPEG-4 standard also defines the following default quantization matrices for intra blocks (independent, non-predicted blocks):

$$\begin{bmatrix} 8 & 17 & 18 & 19 & 21 & 23 & 25 & 27 \\ 17 & 18 & 19 & 21 & 23 & 25 & 27 & 28 \\ 20 & 21 & 22 & 23 & 24 & 26 & 28 & 30 \\ 21 & 22 & 23 & 24 & 26 & 28 & 30 & 32 \\ 22 & 23 & 24 & 26 & 28 & 30 & 32 & 35 \\ 23 & 24 & 26 & 28 & 30 & 32 & 35 & 38 \\ 25 & 26 & 28 & 30 & 32 & 35 & 38 & 41 \\ 27 & 28 & 30 & 32 & 35 & 38 & 41 & 45 \end{bmatrix}$$

And for non-intra blocks (predicted blocks):

$$\begin{bmatrix} 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\ 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 \\ 19 & 20 & 21 & 22 & 23 & 24 & 26 & 27 \\ 20 & 21 & 22 & 23 & 25 & 26 & 27 & 28 \\ 21 & 22 & 23 & 24 & 26 & 27 & 28 & 30 \\ 22 & 23 & 24 & 26 & 27 & 28 & 30 & 31 \\ 23 & 24 & 25 & 27 & 28 & 30 & 31 & 33 \end{bmatrix}$$

The value of QP is not specified in the standard – QP is adjusted constantly from picture to picture and from macroblock to macroblock as a rate-distortion control method.

The inverse quantization simply scales back the values by the quantizer step size:

$$\widehat{\mathbf{T}}_{[u,v]} = \mathbf{T}^{q}_{[u,v]} \cdot \mathrm{QP} \cdot \mathbf{Q}_{[u,v]} \tag{2}$$

Of course, quantization is lossy and the reconstruction will not generally match the original ($\widehat{\mathbf{T}}_{[u,v]} \neq \mathbf{T}_{[u,v]}$).

3.1.3 Motion Search and Motion Compensation

Motion search is performed during encoding only. The motion search algorithm is non-normative, which means the design of the algorithm is left to the engineers to decide. The complexity of the motion search algorithm can be very high, especially for B-pictures where two reference pictures have to be searched.

Motion search is nothing more than finding the best match of the current MB in the reference picture. However, the design of the algorithm is complicated due to limited resources. Usually, one should consider the following components:

- **Search Area:** It is often not practical to search the entire reference picture. The bigger the search area, the more calculation is required. However, it is more likely to find a good match given a big search area. Many algorithms have been developed in order to find the best compromise between search area and computational complexity. A logarithmic search starts with a wide coverage and gradually narrows down the matching area. Some other encoders incorporate a spiral search pattern, where the search radius progressively increases.

- **Cost (Distance) Measure:** In order to rank the candidates in motion search, a "cost" measure is needed. Common measures include MSD (mean square difference – same as MSE), SAD (sum of absolute difference), MAD (mean absolute difference) and SATD (sum of absolute transformed difference). The choice of the cost measure is limited by resources as well. SAD is usually used instead of MSD because it is simpler.

- **Threshold:** It is possible that a given MB might not have a "good" match in the reference picture, e.g., when an object from the current frame does not appear in the reference picture. In this case, it is better to code this MB like those in the I-picture instead of coding the prediction error. This MB is referred to as an *intra* MB in an *inter* picture. The decision of what constitute a "good" match is again the designer's choice. Therefore, a threshold is usually placed on the cost. If the best candidate has a cost greater than the threshold, the current MB will be coded as an intra block.

- **Search Location:** Another challenge in motion search is to decide where to search. A sequential search checks every possible location in a given search area, which is time consuming. A logarithmic search operates iteratively, using the results from the previous iteration to decide which region to search. Other algorithms may also use the motion vectors from previous MBs as references.

After motion search, the matched blocks from the reference picture are put together to form a predicted picture. The difference between the current picture and the predicted picture is called the residual. The residual image is then transformed and quantized, and is passed down the encoder pipeline.