Module 7

VIDEO CODING AND MOTION ESTIMATION

Lesson 20

Basic Building Blocks & Temporal Redundancy

Instructional Objectives

At the end of this lesson, the students should be able to:

1. Name at least five major applications of video compression and coding.
2. Present the block-diagram of a hybrid video codec.
3. Define intra-coded and inter-coded frames.
4. Explain the role of motion estimation in video codecs.
5. Explain the role of motion compensation in video codecs.
6. Define the translational model of motion estimation.
7. Define backward motion estimation.
8. Name two distinct approaches to motion estimation.

20.0 Introduction

So far, we have studied still image coding techniques and the standards therein. We are now going to explore the most challenging area in multimedia communication that is video storage and transmission. A wide range of emerging applications, such as conversational video like video telephone , video conferencing through wired and wireless medium; streaming video, such as video on demand; digital TV/HDTV broadcasting , image / video database services, CD/DVD storage etc, demand significant amount of data compensation. Today, the technology has reached a state of maturity with the availability of coding and compression tools, acceptance of international standards proposed by International Standards Organization (ISO) and International Telecommunications Union (ITU), but research is still on, to achieve further improvements.

In this lesson, we shall introduce the basics of video coding. Most of the theories have been already studied by us in the previous lessons and we shall focus on the exploitation of temporal redundancies, which we could not do for still image compression. Temporal redundancy is exploited through the prediction of current frame using the stored information of the past frames. After presenting the basic building blocks of a video codec, in the remaining part of the present lesson, as well as in Lesson 21, we shall study the methodology for motion estimation. There are two basic approaches to motion estimation, viz, pixel based methods that use optical flow and block based methods. The latter are popular from hardware and software implementation considerations and have been incorporated in all the video coding and multimedia standards.

20.1 Hybrid Video codec

The block-diagram of the basic video encoder and decoder as incorporated in all the video coding standards with some variation and modification is shown in fig 20.1. These codecs are popularly referred to as hybrid codecs, since they use a combination of predictive and transform domain techniques.

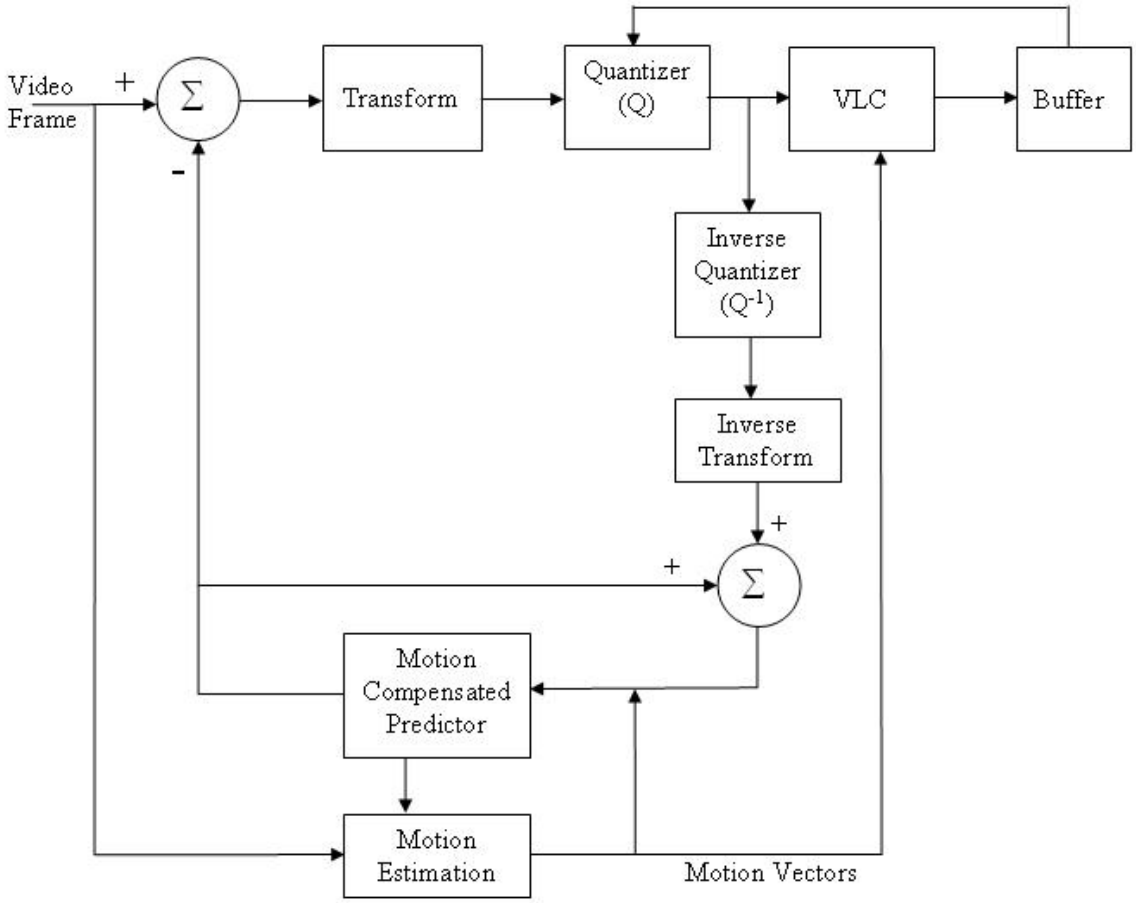

Figure (a)

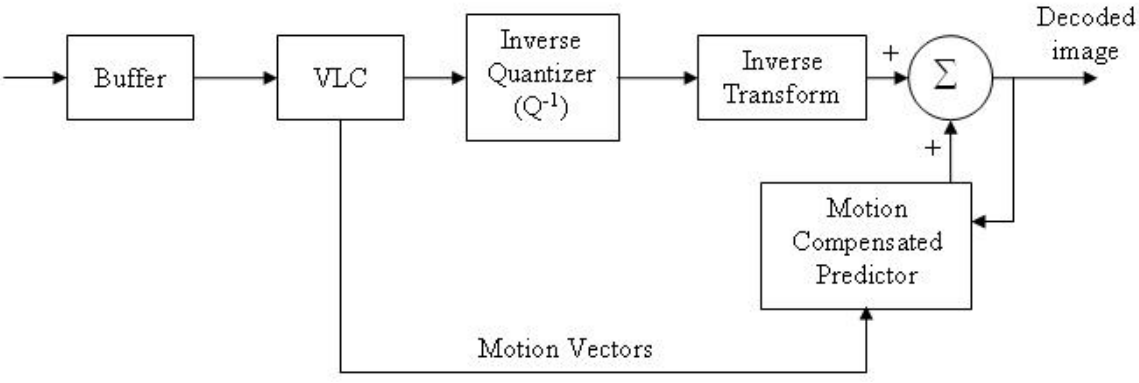

Figure (b)

Fig 20.1: *Block diagram of (a) Video encoder and (b) Video decoder*

The predictive technique is used to temporally predict the current frame from the previous one(s) that must be stored. The temporal prediction is based on the assumption that the consecutive frames in a video sequences exhibit very close similarity, except for the fact that the objects or the parts of a frame in general may get somewhat displaced in position. This assumption is mostly valid except for the frames having significant change of contents. The predicted frame generated by the exploitation of temporal redundancy is subtracted from the incoming video frame, pixel by pixel and the difference is the error image, which will in general exhibit considerable spatial redundancy. The error image goes through transform which is the DCT for MPEG-1, MPEG-2 and ITU-T standard H 261, H 263 etc. The latest ITU-T standard H 264 uses an integer-DCT and MPEG-4 supports wavelet transforms. As in still image codecs, the transformed coefficients are quantized and entropy coded before adding to the bit stream. The encoder has a built in decoder too so as to reconstruct the error frame, which will not be exact, because of the quantizer. The error frame is added to the predicted frame to generate the buffer. The motion estimation block determines the displacement between the current frame and the stored frame. The displacements so computed are applied on the stored frame in the motion compensation unit to generate the predicted frame. We shall discuss about the motion estimation and the motion compensation blocks with further details in Section 20.3.

20.2 Intra-coded and inter-coded frames

The motion estimation and the motion compensation blocks work, only if there is a past frame that is stored. So, question is how do we encode the first frame in a video sequence, for which there is no past frame reference? The answer to this question is fairly straight forward. We treat the first frame of a video sequence like a still image, where only the spatial, i.e the intra-frame redundancy can be exploited. The frames that use only intra-frame redundancy for coding are referred to as the “intra-coded” frames. The first frame of every video sequence is always an intra-coded frame.

From the second frame onwards, both temporal as well as spatial redundancy can be exploited. Since these frames use inter-frame redundancy for data compression, these are referred to as inter-coded frames.

However, it is wrong to think that only the first frame of a video sequence would be “intra-coded” and the rest “inter-coded”. In some of the multimedia standards, intra-coded frames are periodically introduced at regular intervals to prevent accumulation of prediction error over frames.

It is obvious that intra-coded frames would require more bits to encode as compared to inter-coded frames since the temporal redundancy is not exploited in the former.

20.3 Motion estimation and motion compensation

While explaining the block diagram of a generic video codec in Section 20.1, we identified motion estimation and motion compensation as two major blocks which are new additions as compared to the building blocks of an image codec.

The motion estimation block in a video codec computes the displacement between the current frame and a stored past frame that is used as the reference. Usually the immediate past frame is considered to be the reference. More recent video coding standards, such as the H.264 offer flexibility in selecting the references frames and their combinations can be chosen. Fig. 20.2 illustrates the basic philosophy of motion estimation. We consider a pixel belonging to the current frame, in association with its neighborhood as the candidates and then determine its best matching position in the references frame. The difference in position between the candidates and its match in the reference frame is defined as the *displacement vector* or more commonly, the *motion vector*. It is called a vector since it has both horizontal and vertical components of displacement. We shall offer a more formal treatment to motion estimation in the next sections.

After determining the motion vectors one can predict the current frame by applying the displacements corresponding to the motion vectors on the reference frame. This is the role of the motion compensation unit. The motion compensation unit therefore composes how the current frame should have looked if corresponding displacements were applied at different regions of the reference frame.

20.4 Translational model of motion estimation

In the past section, we had inherently made an assumption that the candidate pixels, along with its surroundings in the current frame undergo displacements with respect to the reference of size and orientation of the region considered. In other words, we make an assumption that the displacement is purely translational and scaling, rotation or any three dimensional deformation in general can be neglected. This assumption is not strictly valid, since we capture 3-D scenes through the camera and objects do have more degrees of freedom than just the translational one. However, the assumptions are still reasonable, considering the practical movements of the objects over one frame and this makes our computations much simpler.

Let $s(n_1, n_2, k)$ be a pixel at spatial coordinate (n_1, n_2) in an integer grid , corresponding to the frame-*k*. Thus, $(n_1, n_2, k) \in \Lambda^3$, i.e. the three-dimensional integer space. We choose *k* as the current frame and a past frame (*k-l*) having *l* >0 as the reference frame. If (d_1, d_2) is the motion vector corresponding to the position (n_1, n_2), the translational model assumes

$$s(n_1, n_2, k) = s(n_1 - d_1, n_2 - d_2, k - l) \quad \text{.....(20.1)}$$

for all pixels in the neighborhood of (n_1, n_2) for spatial consistency. The restriction of translational motion is usually imposed on block based motion estimation, but there is no such restriction on optical-flow based methods.

20.5 Backward motion estimation

The motion estimation that we have discussed in Section-20.3 and Section 20.4 is essentially backward motion estimation, since the current frame is considered as the candidate frame and the reference frame on which the motion vectors are searched is a past frame, that is, the search is backward. Backward motion estimation leads to forward motion prediction.

Backward motion estimation, illustrated in fig 20.2 stipulates that in equation (20.1), *l* is always greater than zero.

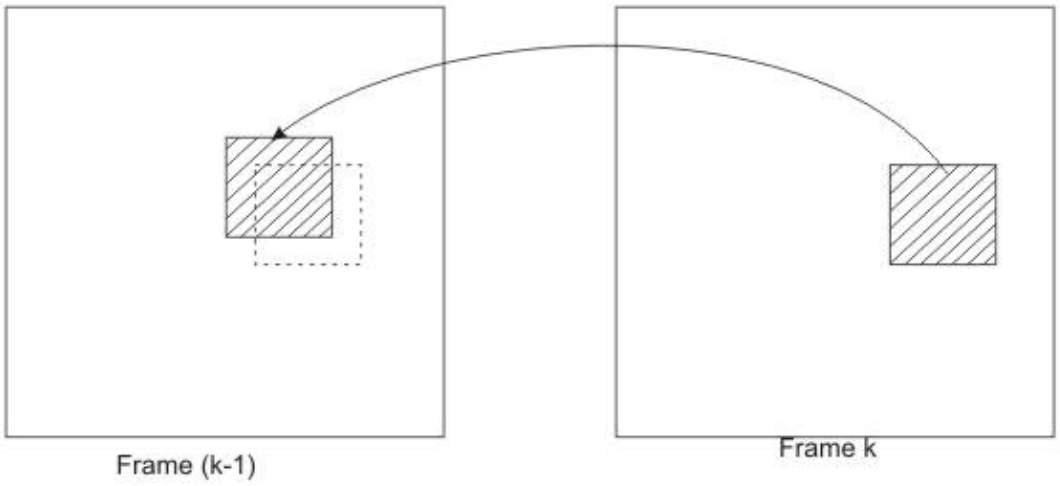

Fig. 20.2 Backward motion estimation with frame k as the current frame and frame (k-1) as the reference frame

20.6 Forward motion estimation

It is just the opposite of backward motion estimation. Here, the search for motion vectors is carried out on a frame that appears later than the candidates frame in temporal ordering. In other words, the search is “forward”. Forward motion estimation leads to backward motion prediction.

Forward motion estimation, illustrated in fig 20.3 stipulates that in equation (20.1), *l* is always less than zero.

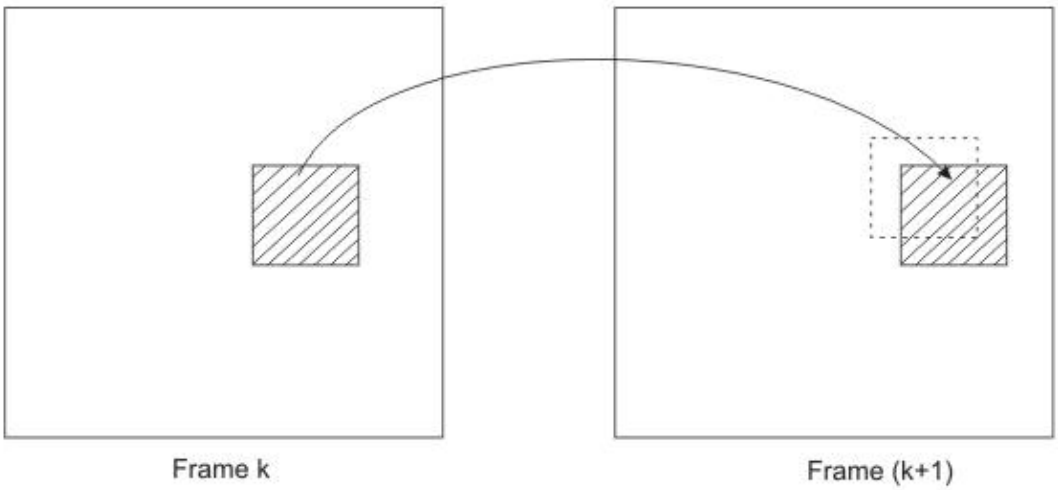

Fig. 20.3 Forward motion estimation with frame k as the current frame and frame (k+1) as the future reference frame

It may appear that forward motion estimation is unusual, since one requires future frames to predict the candidate frame. However, this is not unusual, since the candidate frame, for which the motion vector is being sought is not necessarily the current, that is the most recent frame. It is possible to store more than one frame and use one of the past frames as a candidate frame that uses another frame, appearing later in the temporal order as a reference.

Forward motion estimation (or backward motion compensation) is supported under the MPEG 1 & 2 standards, in addition to the conventional backward motion estimation. The standard also supports bi-directional motion compensation in which the candidate frame is predicted from a past reference as well as a future reference frame with respect to the candidates frame. This will be explained in details when we discuss about the video coding standards in subsequent lessons.

20.7 Basic approaches to motion estimation

There exists two basic approaches to motion estimation –

a) Pixel based motion estimation

b) Block-based motion estimation.

The pixel based motion estimation approach seeks to determine motion vectors for every pixel in the image. This is also referred to as the optical flow method, which works on the fundamental assumption of brightness constancy, that is the intensity of a pixel remains constant, when it is displaced. However, no unique match for a pixel in the reference frame is found in the direction normal to the intensity gradient. It is for this reason that an additional constraint is also introduced in terms of the smoothness of velocity (or displacement) vectors in the

neighborhood. The *smoothness constraint* makes the algorithm interactive and requires excessively large computation time, making it unsuitable for practical and real time implementation.

An alternative and faster approach is the block based motion estimation. In this method, the candidates frame is divided into non-overlapping blocks (of size 16 x 16, or 8 x8 or even 4 x 4 pixels in the recent standards) and for each such candidate block, the best motion vector is determined in the reference frame. Here, a single motion vector is computed for the entire block, whereby we make an inherent assumption that the entire block undergoes translational motion. This assumption is reasonably valid, except for the object boundaries and smaller block size leads to better motion estimation and compression.

Block based motion estimation is accepted in all the video coding standards proposed till date. It is easy to implement in hardware and real time motion estimation and prediction is possible.

In the next lesson, we shall discuss the details of block based motion estimation and the associated algorithms.