Module 7

VIDEO CODING AND MOTION ESTIMATION

Lesson 21
Block based motion estimation algorithms

Lesson Objectives

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

1. Name and define the matching criteria for block motion estimation.
2. Determine the computational complexity of each matching criterion
3. Explain full search block motion (FSBM) estimation.
4. Determine the computational complexity in FSBM.
5. State the fundamental assumption of quick search strategy.
6. Explain 2-D logarithmic search strategy.
7. Determine the computational complexity of 2-D logarithmic search.

21.0 Introduction

In lesson 20, we had introduced video codecs and discussed the role of motion estimation block. We had pointed out that block-based motion estimation is preferred over pixel based methods in practical implementations. In this lesson we are going to cover different popular algorithms on motion estimation, proposed till date. The simplest, but the most time consuming one is the full search block motion (FSBM) estimation that exhaustively searches for the best motion vector of each block within a specified search range. However searching in every candidate position increases the computational complexity of the algorithm. Several quick and efficient search methodologies, such as 2-D Logarithmic search, three steps search (TSS), new three steps search (NTSS), cross search, gradient descent search, four step search etc. have been proposed in recent times. These algorithms are suboptimal in performance, as compared to FSBM, but significantly reduce the computational complexities.

In this lesson, we shall first define different matching criteria used for block-based motion estimation. This will be followed by FSBM and the 2-D Logarithmic search. The other quick search algorithms will be presented in lesson-22.

21.1 Matching Criteria for block-based motion estimation

Three popular matching criteria used for block-based motion estimation are:-

a) Mean of squarred error (MSE)

b) Mean of absolute difference (MAD)

c) Matching pel count (MPC)

To implement the block motion estimation, the candidate video frame is partitioned into a set of non overlapping blocks and the motion vector is to be determined for each such candidate block with respect to the reference.

For each of these criteria, square block of size *N* x *N* pixels is considered. The intensity value of the pixel at coordinate (n_1, n_2) in the frame–*k* is given by $S(n_1, n_2, k)$, where $0 \le n_1, n_2 \le N-1$.

The frame–*k* is referred to as the candidate frame and the block of pixels defined above is the candidates block. With these definitions, we now define the matching criteria.

21.1.1 MSE Criterion

Considering (*k-l*) as the past references frame $(l > 0)$ for backward motion estimation, the mean square error of a block of pixels computed at a displacement (I,j) in the reference frame is given by

$$MSE(i, j) = \frac{1}{N^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} [s(n_1, n_2, k) - s(n_1 + i, n_2 + j, k - l)]^2 \quad \text{.....(21.1)}$$

The physical significance of the above equation should be well understood. We consider a block of pixels of size *N* x *N* in the reference frame, at a displacement of (i, j), where *i* and *j* are integers with respect to the candidate block position.

The MSE is computed for each displacement position (i, j) within a specified search range in the reference image and the displacement that gives the minimum value of MSE is the displacement vector which is more commonly known as motion vector and is given by

$$[d_1, d_2] = \arg \min_{i,j} [MSE(i, j)] \quad \text{.....(21.2)}$$

The MSE criterion defined in equation (21.1) requires computation of N^2 subtractions, N^2 multiplications (squaring) and $(N^2 - 1)$ additions for each candidate block at each search position. This is computationally costly and a simpler matching criterion, as defined below is often preferred over the MSE criterion.

21.1.2. MAD criterion:

Like the MSE criterion, the mean of absolute difference (MAD) too makes the error values as positive, but instead of summing up the squared differences, the absolute differences are summed up. The MAD measure at displacement (i, j) is defined as

$$MAD(i,j)=\frac{1}{N^2}\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}\left[s(n_1,n_2,k)-s(n_1+i,n_2+j,k-l)\right] \quad \text{.....(21,3)}$$

The motion vector is determined in a manner similar to that for MSE as

$$[d_1,d_2]=\arg\min_{i,j}[MAD(i,j)] \quad \text{.....} \quad (21.4)$$

The MAD criterion requires computations of N^2 subtractions with absolute values and N^2 additions for each candidate block at each search position. The absence of multiplications makes this criterion computationally more attractive and facilitates easier hardware implementation.

21.1.2 MPC criterion :

In this criterion, the pixels of the candidates block *B* are compared with the corresponding pixels in the block with displacement (i,j) in the reference frame and those which are less than a specified threshold, i.e., closely matched are counted. The count for matching and the displacement (i,j) for which the count is maximum correspond to the motion vector. We define a binary valued function $count(n_1,n_2)\forall(n_1,n_2)\in B$ as

$$count(n_1,n_2)=\begin{cases}1 & \text{if } |s(n_1,n_2,k)-s(n_1+i,n_2+j,k-l)|\le\theta\\ 0 & \text{otherwise}\end{cases} \quad \text{.....(21.5)}$$

where, θ is a pre-determined threshold. The matching pel count (MPC) at displacement (i,j) is defined as the accumulated value of matched pixels as given by

$$MPC(i,j)=\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}count(n_1,n_2) \quad \text{.....(21.6)}$$

$$[d_1,d_2]=\arg\max_{i,j}[MPC(i,j)]$$

21.2 Full-search block motion (FSBM) estimation

In FSBM estimation, the search for the candidate block is exhaustively carried out in all possible positions within a specified search range $\pm w$ in the reference frame. Fig 21.1 illustrates this principle

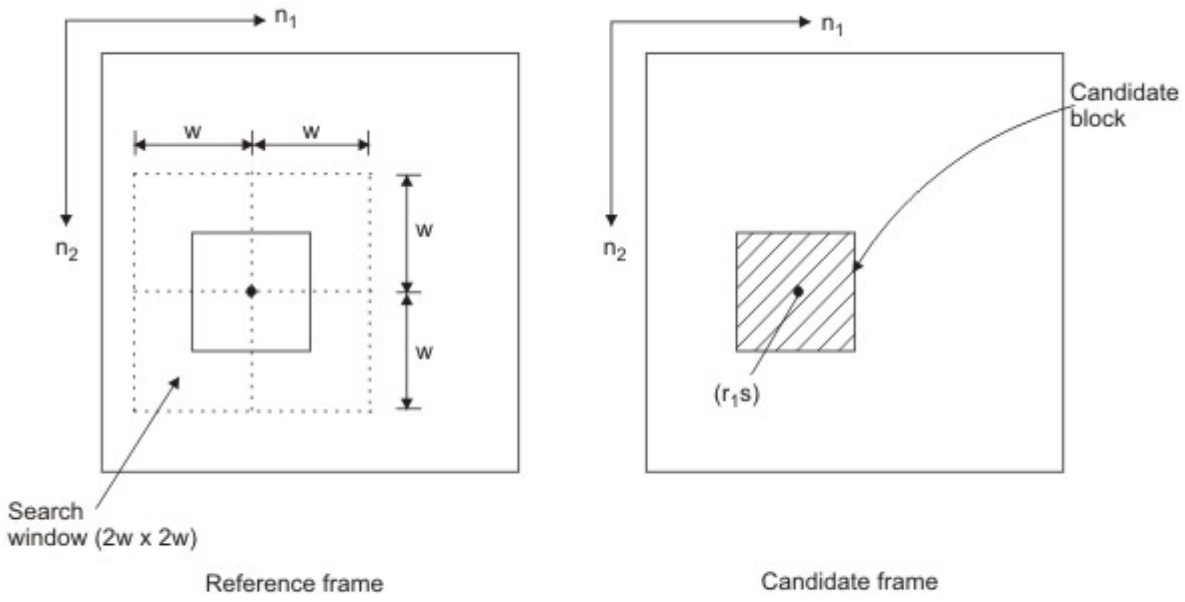

Figure 21.1 Full-search block motion estimation

We consider a block of *N* x *N* pixels from the candidates frame at the coordinate position *(r,s)* as shown. We then consider a search window having a range $\pm w$ in both n_1 and n_2 directions in the references frame, as shown.

For each of the $(2w+1)^2$ search position (including the current row and the current column of the reference frame), the candidate block is compared with a block of size *N* x *N* pixels, according to one of the matching criteria discussed in section 21.1 and the best matching block, along with the motion vector is determined only after all the $(2w+1)^2$ search position are exhaustively explored.

The FSBM is optimal in the sense that if the search range is correctly defined, it is guaranteed to determine the best matching position. However, it is highly computational intensive. For each matching position, we require $O(N^2)$ computations (additions, subtractions, multiplications etc) and since there are $(2w+1)^2$ search positions, the number of computations for each matching criterion is given by Table 21.1

Table 21.1 Computational complexity of FSBM

Matching Criterion	Computations		
	Additions/ Subtractions	Multiplication	Comparison
MSE	$(2N^2-1)(2w+1)^2$	$N^2(2w+1)^2$	$(2w+1)^2$
MAD	$(2N^2-1)(2w+1)^2$	---	$(2w+1)^2$
MPC	$(2N^2-1)(2w+1)^2$	----	$N^2(2w+1)^2$

We therefore conclude that FSBM requires large number of computations. Say, we take $w=7$ pixels, measuring that the best matching position exists within a displacement of $\pm$ 7 pixels from he current block position, we require 15 x 15=225 search positions. For real time implementation, quick and efficient search strategies were explored.

It may be noted that such quick search techniques do not make exhaustive search within the search within the search area and can at best be sub-optimal.

21.3 Quick and efficient search strategies

The quick and efficient search strategies work with a heuristic assumptions that as we move away from the minimum distortion position, the distortion function (given by the *MSE* or the *MAD* criterion) monotonically increases in any of the four quadrants that can be formed with the best search position as the origin.

Some of the popular and efficient search strategies are explained below.

21.3.1 2-D Logarithmic search :

In this search strategy originally proposed by Jain and Jain [1], an initial search window is considered, with an assumed best position as its centre. The initial guess of the best position may be taken as the position of the candidate block with respect to the candidate frame. Instead of searching in all possible displacement positions within the search window, search is done only in five locations which contains the centre and the mid points between the centre and the four edges of the search window.

The minimum distortion position out of these five search positions is taken as the new guess of the best position and is considered as the centre of the search window for the next step, in which the search area is reduced by a factor of two. This algorithm is iteratively continued until the search window is reduced to 3 x 3 size. In the final step, all the nine locations are searched and the location corresponding to the minimum distortion is the best matching position. The corresponding displacement of the best matching position with respect to the candidate block coordinates gives the motion vector.

The 2-D logarithmic search strategy is illustrated in Fig.21.2. The numbers indicate search steps, those circled show the optimum position for that search step and the * shows the final optimum position.

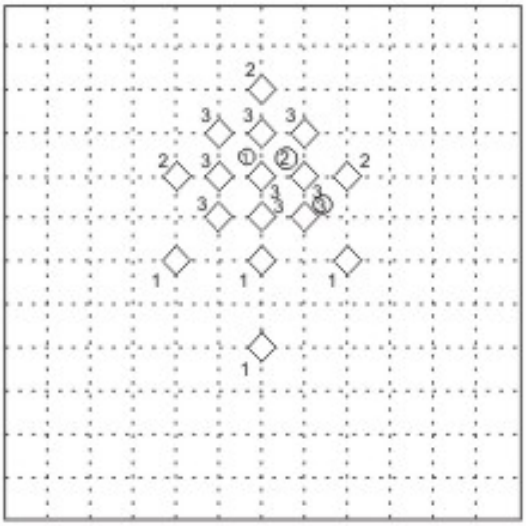

Figure 21.2 2-D Logarithmic search procedure

Source: http://nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Multimedia%20Processing/pdf/ssg_m7l21.pdf