Digital Voice and Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

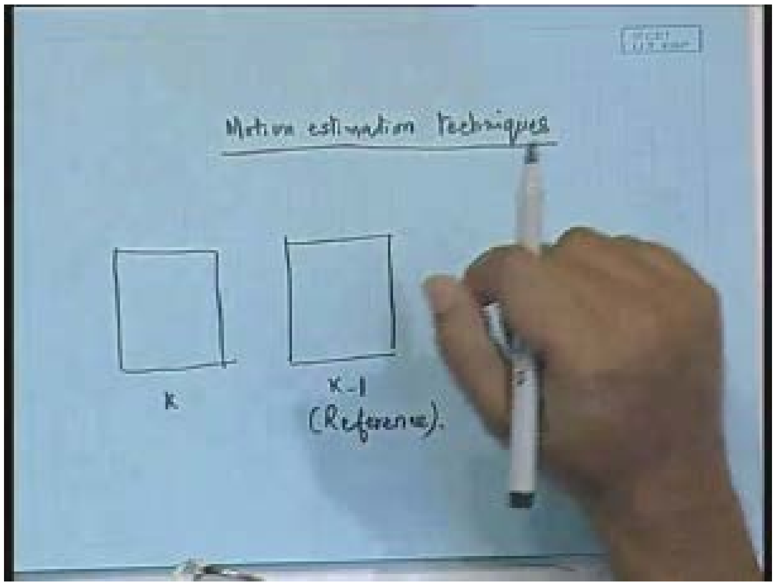
Indian Institute of Technology, Kharagpur

Lecture - 24

Motion Estimate Techniques

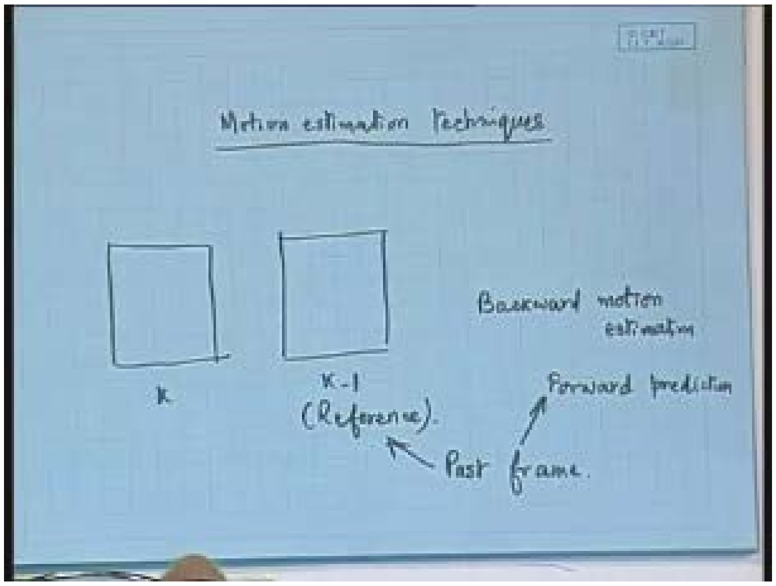
....................discuss about the motion estimation techniques. Now, the basic concepts of motion estimation I have already mentioned in the last class when I had introduced the basic concepts behind video coding. What we had discussed last time is that, if we have to encode a frame and we like to exploit the temporal redundancy in order to achieve significant amount of compression in that case what we do is that if we take our candidate frame to be the frame number k, let us say some frame number k which is the candidate frame, in that case the candidate frame k could be encoded by one of the past frames. By using one of the past frames let us say, the immediate past frame and that immediate past frame we should be designated as that should be designated as k minus 1. So k minus 1th frame which is used as the reference is going to predict what frame number k is going to be. Hence, after finding out the motion vectors, it will apply the motion vectors on to this reference frame and predict k and the difference between the incoming frame k and the predicted frame k that difference has to be encoded that is what we said.

(Refer Slide Time: 2:35)



Hence, basically what we have is that a past frame; what we are using as a reference is a past frame and on the past frame on the basis of past frame we are predicting what the future is. So basically whenever it is estimating the motion it is going back in time in order to find out the motion vectors and when it is a question of compensation that means to say predicting what it is going to be in that case the prediction is going to be a forward prediction. So whenever we are using a past frame as a reference then it leads to a forward prediction. So you can say that the process what we have described basically means that there is a backward motion estimation; why backward is because we are going back in time, we are considering the past frame for motion estimation so we are calling that as the backward motion estimation and whenever we are using backward motion estimation it leads to forward prediction or forward motion compensation whatever we would like to call.

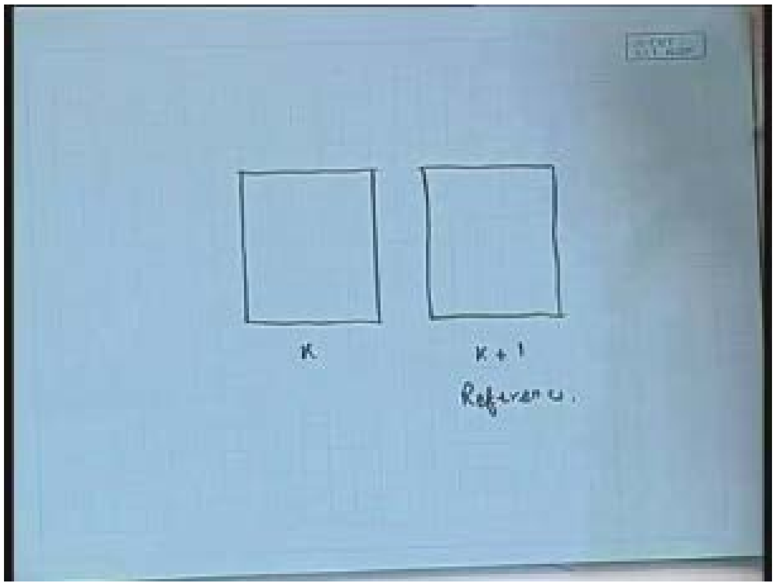
(Refer Slide Time: 3:56)



Now we can put the concept in the opposite way also; although it would look very puzzling but let us say that why do we use a past frame for a reference why do not we use it future frame as a reference. Looks little awkward because if we are going to encode frame k now, in that case what I have just now suggested that we should be using some future frame as a reference; let us say the next future frame; the next future frame would be k plus 1th frame. So k plus 1th frame we are going to use as reference and with respect to that we are going to predict k looks little awkward because we have not yet received the future frame.

The frame k is as if like our incoming frame k and then we are going to predict frame k on the basis of something which is in the future which has not come yet. Although the concept looks somewhat puzzling but let us say that if at this moment of time some frame is going to be encoded it does not necessarily mean that that is the coming that is the incoming frame coming out of the real time source. What I mean to say is that at this moment of time while the camera may be capturing frame k plus n let us say I can encode frame k. That means to say that I am only introducing some kind of a delay. What we can do is to have a prior storage of the frames. So, if we have some prior storage of frames; in this case let us say that if n frames are kept stored and while frame k plus n is being processed by the camera while frame k plus n is being capturing by the camera we have frame number k already available in the memory for encoding so what we are going to do is that frame k is there in the memory frame k plus 1 that is also there in the memory. Therefore, in such case it is possible that we can predict the frame number k by using some frames beyond k as a reference. So k could be the frame to be encoded; we can still call that as current frame no problem; as far as the encoding is concerned it is the current frame to be encoded; may be that with respect to camera it is delayed but frame k............. the encoder treats frame k as the current frame and k plus 1 as a reference frame which is a future reference frame but that is also stored in the memory.

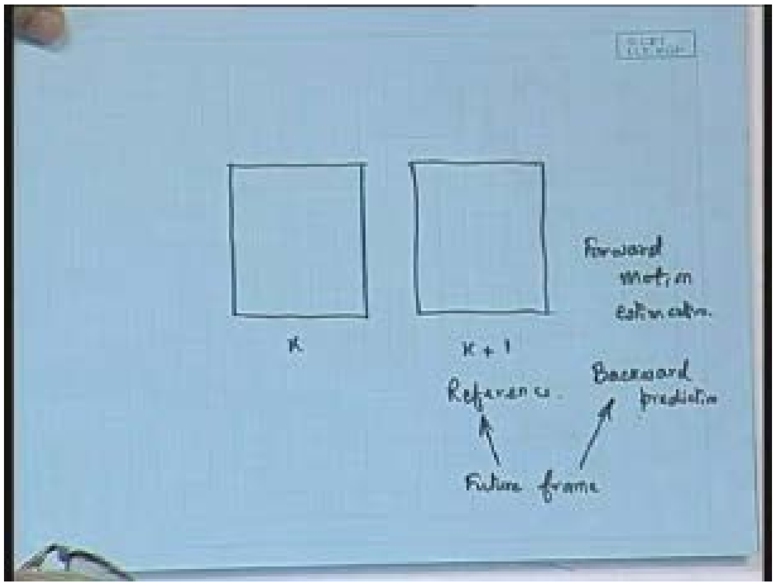
(Refer Slide Time: 7:42)



Hence, now what we are going to do is that we are again going to predict the motions. Or what we are going to do is that, in a very similarly manner what we are going to do is to subdivide the frame k into a number of non-overlapping blocks and for each of these blocks we are going to find out the motion vector which come out of the motion estimation with frame k plus 1 as the reference.

Therefore, in this case what we are doing is that we are using the future frame as a reference; with respect to frame k we are using future frame as a reference which means to say that the prediction that it is doing is for a frame which is older which means to say that this is a backward prediction, this is a backward prediction and the motion estimation process in this case is a forward motion estimation why; because frame k is the candidate frame and it is finding out the motion vector by considering frame k plus 1 which is forward in time so it is a forward motion estimation that leads to a backward prediction.

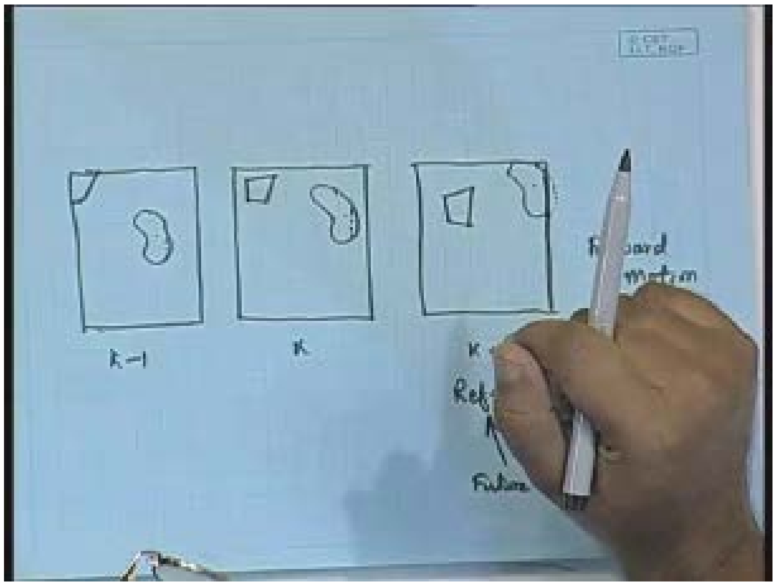
(Refer Slide Time: 8:38)



Therefore, the first case, the case what we discussed earlier that was a forward prediction and what I have described just now is the case of backward prediction. Now why is forward prediction required, when is forward prediction required, when is backward prediction required.

You see, what we are doing is that we are dealing with moving objects. Let us say that we draw three frames. There is frame k, there is frame k plus 1, there is also frame k minus 1 we have got three frames and let us say that there is some object let us say this is the object which is at some position in frame k minus 1 (Refer Slide Time: 9:24). Now, in frame k plus k it has moved by some extent, some motion vector is there so by that it has moved and in the frame k plus 1 it has moved to further so let us say that it has come to this position. So what you can clearly see is that in frame k plus 1 the part of the object the object boundary would may be would have been like this so the part of object boundary lies outside the frame so it is not visible to us; what is visible to us is only what we have within this rectangular box so this is beyond the frame size so there is some amount of occlusion which is taken which has taken place means that this part of the object which was visible in frame k minus 1 (Refer Slide Time: 10:18) and which is visible in frame k is not visible in frame k plus 1; the opposite also could have happened like I might have got some object like this in frame k minus 1 and then in frame k plus 1 the object may be like this in frame k the object may be like this and in frame k plus 1 the object may be like this. So in this case what is happening is that this object is partly occluded in frame k minus 1 but it is unoccluded in frame k and frame k plus 1.

(Refer Slide Time: 10:57)



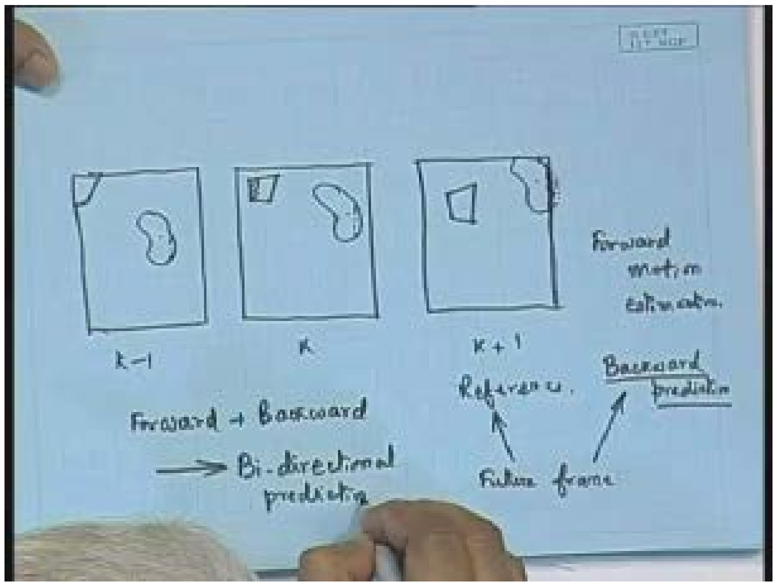
So how the object moves is not something that we can really tell about this because these are physical objects so they may move they may be out of screen and slowly come into the screen or they may be inside the screen and go out of the screen either of these two cases can happen. So, when we have some occlusion let us say that when this is the scenario in that case we cannot have any prediction for this region of the object why because simply this region is not available in the previous frame (Refer Slide Time: 11:38) whereas in this case if we do some forward prediction the forward prediction or the backward motion estimation would be problematic because in this case we have only part of this object so we will not be having any part like this which is not visible those parts will not have any correspondence in the previous frame.

Now, therefore, the best combination could be that, if we employed, I mean, since we are having the idea of storing the frame in the memory and then use it for encoding in that case we should be able to use both past frame as well as the future frame for our motion prediction. We can use both the past and the present for prediction and in that case what we are doing is that we are employing both forward prediction and backward prediction.

Now when we are using both forward as well as backward prediction then forward and backward prediction combination that leads to a bidirectional prediction. Now, why should people use bidirectional prediction? The reason behind this is that; of course whenever you are having two reference frames if you commit any mistake in estimating the motion with respect to the past frame may be the same amount of mistake you are not going to have while having the backward motion prediction. If you have error in forward prediction you may not have the similar error in the backward prediction or vice versa. So there is always something; I mean, averaging always gives you better estimate because one estimate could be wrong and that may be corrected by another estimate.

And not only that, the question of the occlusion also could be there that some object may be occluded in the past frame and some object may be occluded in the future frame. But whenever you are using both forward and backward prediction in that case you can decide that whether to take forward as the reference or whether to use the backward as the reference or whether to use both as the reference so bidirectional prediction also could be employed.

(Refer Slide Time: 13:06)



Now this leads to some important concept about the picture encoding picture means that video frame file I mean to say; that when the video frames are continuously arriving one can have a forward prediction, one can have a backward prediction, one can have a bidirectional prediction.

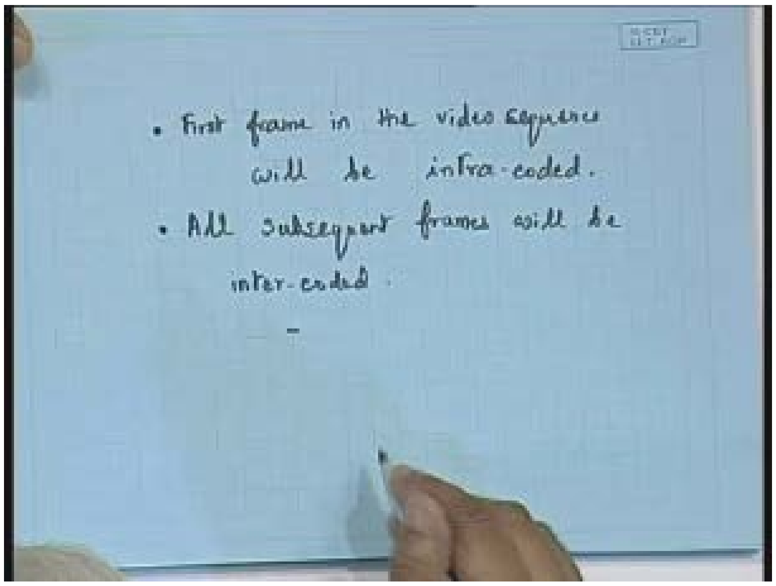
Now, in the video coding standards all these mechanisms are employed. But before I go into the aspect of forward and the backward predictions and using it in the encoding of the frames let us address a very fundamental question. Let us say that a video sequence is starting. So we have started a video sequence, we have started encoding the video sequence and we have the first frame to be encoded. Now the first frame does not have any past reference so what should be our starting point? We do not have any past reference and because we do not have any past reference there is no frame that has been stored also it is just coming for the first time, we have obtained the first frame from the camera and we would like to encode it and in absence of any reference the only mechanism that we can adopt in order to encode the first frame is to treat the first frame just like a still image.

Just like the way we said while still image while talking about the still image compression that means to say that you can only exploit the spatial redundancy you can adopt techniques like the

DCT or the wavelet in order to encode the first frame. The first frame is not having any inter-frame prediction. By inter I mean to say that using another frame for prediction in between the frames. So we are not using inter-frame so I can only say that the first frame will be intra coded so the first frame in the video sequence first frame in the video sequence will be intra-coded.

Subsequent to the first frame we do not have problems. Because when the second frame comes I can use the first frame as the past reference and then if I store the second frame then the third frame is the incoming frame or rather to say that second frame has come I have stored it, third frame has come I have stored it, fourth frame is coming and I would like to encode the second frame in that case I can use, for the second frame I can use the first frame as the reference, I can use the third frame as the reference.

(Refer Slide Time: 18:49)



- First frame in the video sequence will be intra-coded.
- All subsequent frames will be inter-coded.

Now it is not essential that the prediction will be done only by one frame forward or one frame backward. Prediction by one frame is only as an example that I said. But you can predict what is going to be three frames ahead or five frames ahead or you can predict with respect to what was three frames behind, four frames behind, five frames behind like that. So when k is the candidate frame we can have k plus n as a reference, we can use k minus n as a reference where n is a quantity which could be an integer greater than zero. In such case.............. so all subsequent frames would be inter-coded all subsequent frames will be inter-coded.

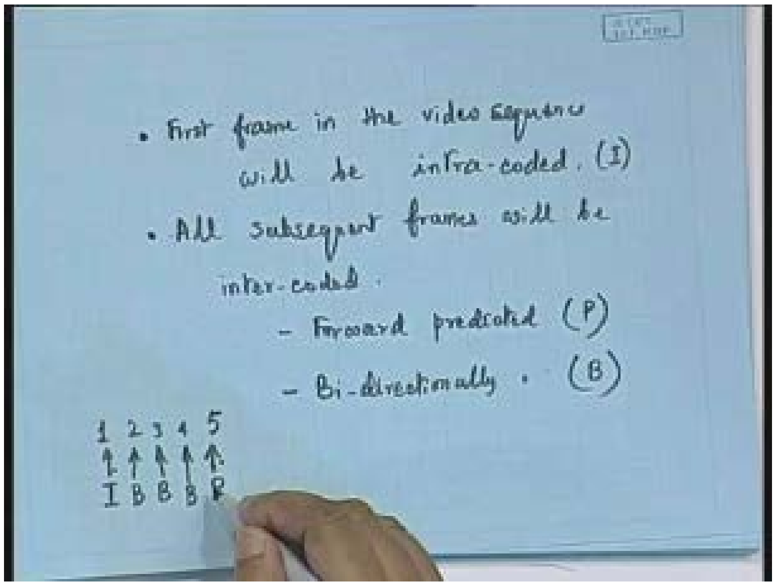
Now, whenever we are doing inter-frame coding in that case one can use forward prediction or backward prediction or bidirectional prediction. So whenever it is forward predicted whenever it is forward predicted in that case means using the past reference forward predicted frames are referred to as P frames, intra-coded frames are refer to as I frames and if we are doing bidirectional prediction; so if the frame is bidirectionally predicted then we are going to call it as B frames. So we have I frames, P frames and B frames which are present in the video sequence.

Now what is the ordering that one has to follow?
See, the first frame in a sequence has to be an I frame. Now let us say that we can use this I frame in order to predict few frames ahead. Let us say second, third, fourth, fifth; fifth frame we are having a forward prediction. That means to say that whenever I am encoding the fifth frame I already have the first frame stored in the memory, so by comparing the fifth frame with the first frame I can obtain that what the motion vector is going to be and I can apply a forward motion prediction on the frame number 1 to predict what frame number 5 is going to be. So frame number 5 would be a P frame. So this is frame number 1, this is frame number 5 (Refer Slide Time: 20:32) which is a P frame.

Now interestingly I have not yet encoded frame number 2, 3 and 4 which are sandwiched in between frame number 1 and frame number 5. Now, after predicting frame number 5 what we can do is that we can use both frame number 1 and frame number 5 as the reference. If we have to encode a frame in between let us say frame number 2; when I have to encode frame number 2 I can use 1 as the reference, I can use 5 as the reference which means to say that I can have a bidirectional prediction. Also, for frame number 3 and 4 one can adopt bidirectional prediction so frame 2, 3 and 4 have to be bidirectionally predicted.

So I have a I frame, I have a P frame here and in between the I frame and P frame I have three B frames which are sandwiched in between. Three is just an example; it could have been two B frames it could have been four B frames anything like that.

(Refer Slide Time: 21:51)



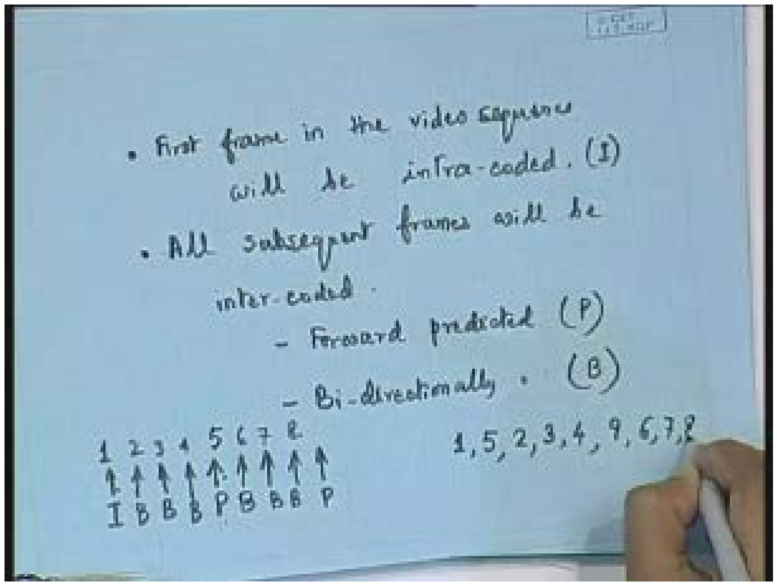
- First frame in the video sequence will be intra-coded. (I)
- All subsequent frames will be inter-coded.
  - Forward predicted (P)
  - Bi-directionally. (B)

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ I & B & B & B & P \end{array}$$

Now, because fifth frame is available now I can predict by a very similar mechanism; I can predict what frame number 9 is going to be. so frame number 9 could be used as a P frame and because I have both 5 and 9 it is possible to bidirectionally predict frame 6, frame 7 and frame 8 so that all these frames 6, 7 and 8 would be the B frames. This is the sequence in which the frames are getting encoded: I B B B P again B B B P like that. this is the sequence in which So this is the

This is the total scenario that we would like to encode frame 1 as I, 2 as B, 3 as B, 4 as B, 5 as P, again 6 as B, 7 as B like that. But whenever we are encoding we are encoding not in the correct order because first we have to encode frame number 1, then we have to encode frame number 5 and only when 1 and 5 are encoded it is possible for me to encode 2, 3 and 4 and then I have to encode 9 and only when both 5 and 9 are encoded I can encode 6, 7 and 8 so there is a juggling in the ordering of the encoding that is being done and that has to be there if we want this prediction to be done like this.

(Refer Slide Time: 23:26)



- First frame in the video sequence will be intra-coded. (I)
- All subsequent frames will be inter-coded.
  - Forward predicted (P)
  - Bi-directionally. (B)

1 2 3 4 5 6 7 8

I B B B P B B B P

1, 5, 2, 3, 4, 9, 6, 7, 8

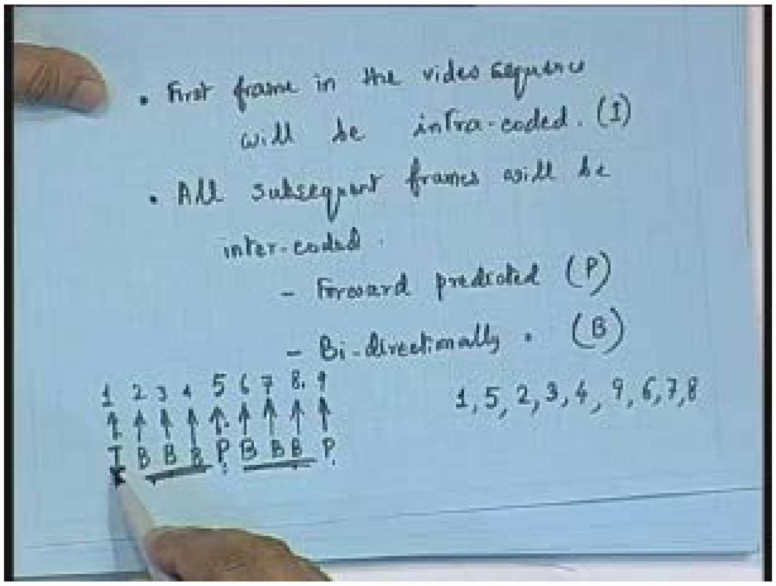
Now what is the advantage in having a prediction like this?

You see, first frame is definitely without any past reference. So in absence of any temporal redundancy it is quite expected that we are consuming a large number of bits in order to encode. So, if we have.......... let us say that for a part of the video if I have this many this much as my available bit budget then it is possible that i use a bulk of it for encoding the I frame; bulk of my big budget would be encoded for I frame. Let us say that I encode up to frame number 13 let us say; from frame number 1 to frame number 13 I encode so I frame would contain the maximum of number of bits would consume maximum number of bits and followed by I the P frames would consume some amount of bits but definitely significantly less as compared to the I frames but the B frames would consume very little bits because as you can see that number of B frames are much more than the number of P frames or the number of I frames so the B frames would consume much less bits so only this much (Refer Slide Time: 25:07) whereas P takes this much; four P frames takes this much and one I frame could be taking this much, this is just a fictitious example that this is the way it is. That means to say that we would like to have very few I frames.

I have so far considered that only the first frame is an I frame and we should have as many B frames as possible. But let us have a look at some practical consideration. The practical consideration is that, whenever we are having whenever we are estimating the motion then we are making some prediction error, and; of course we are not terribly worried about prediction error because as I said that the prediction error would be encoded; the differential which is the prediction error that will be encoded by exploiting the spatial redundancy and that information goes into the channel so that the decoder not only gets the motion vectors but decoder also gets what the prediction error is going to be and then the reconstruction could be done. So, some amount of error we are not terribly worried but whenever we are using.......I mean, to begin with there may not be a problem that frame 1 to frame 2 we make some prediction error; now frame 3 would be predicted by frame 2 or frame 4 would be predicted by...... I mean whatever, I mean, as we proceed in time then our sanctity of the reference that also would degrades.

Like say for example, between frame number 1 and 5 I have used, I frame and P frame as reference and I generated this B‘s (Refer Slide Time: 27:13). But between frame 5 and 9 I take 5 and 9 as reference and I generated these B’s. Now, would these three B’s be better or these three B’s be better? Definitely these three B’s (Refer Slide Time: 27:23) why I am telling this is because here this is the I frame so this is without any prediction error.

(Refer Slide Time: 27:33)



- First frame in the video sequence will be intra-coded. (I)
- All subsequent frames will be inter-coded.
  - Forward predicted (P)
  - Bi-directionally. (B)

1 2 3 4 5 6 7 8 9

I B B B P B B B P

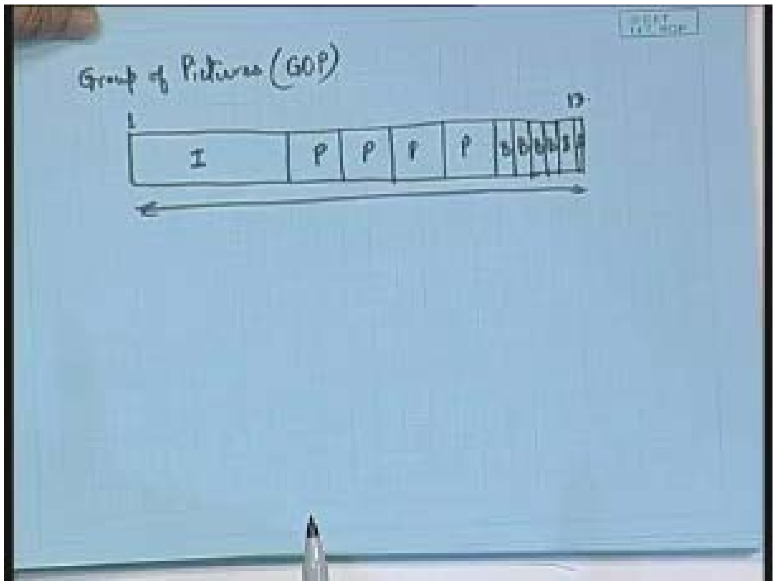
1, 5, 2, 3, 4, 9, 6, 7, 8

Now, when I am predicting frame number 9 by now I have accumulated some amount of prediction error so the sanctity of my reference has degraded and if my reference itself is not having the proper sanctity in that case the in between sandwiched frame that I am generating by treating the frame number 5 and 9 as reference that also will have some amount of error. So if we proceed in this manner it is possible that after sufficient time has elapsed let us say that after hundred frames we may find that our prediction is awfully bad especially whenever we are having rapidly moving objects.

So, in such kind of cases what we should do?
In such cases the mechanism that is employed is that we should forcibly, I mean, after some predetermine time we should forcibly introduce some I frame because just to ensure that the predict the that the prediction error does not accumulate beyond some stage. That is the reason why we introduce some I frames forcibly so this is called as the forced updating and in such kind of case what we can have is that we can divide the picture structure as a group of frames, divide the video in a structure like this that we can form a structure which we can call as a group of frames or group of pictures because group of pictures is a terminology that the standard makers

use so it is called as the GOP. So one GOP configuration; so one particular GOP may have a bit distribution like this.

(Refer Slide Time: 29:43)



Now, one GOP would be followed by the next GOP. What to say is that; if let us say that I take......... a GOP consists of thirteen frames so N is equal to 13 so when the fourteenth frame comes again the next GOP starts and the next group of pictures start with another I frame. I explain to you the concept behind the I frame, P frame and B frame and these terminologies have been used in the video coding standards.

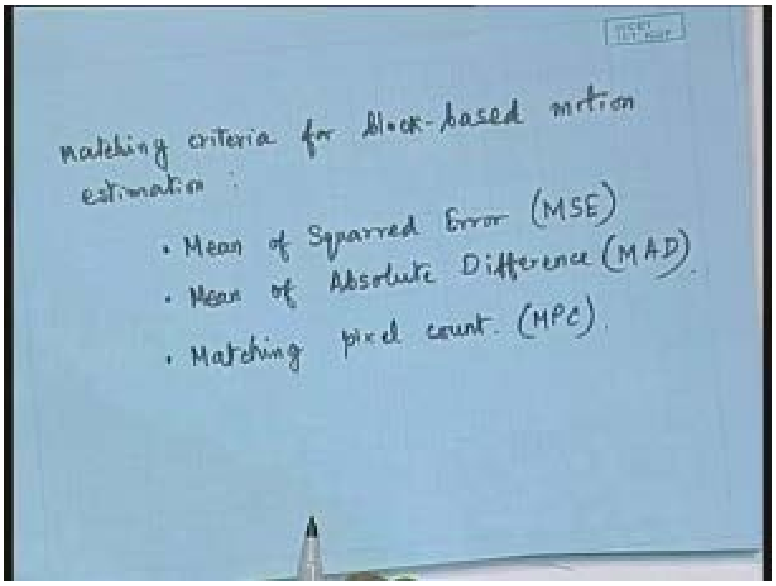
I will come to the discussion of the video coding standards very shortly within next one or two classes only. But before that what I wanted to say a bit about is that how this motion estimation is being done. We have been talking about motion estimation. But the motion estimation is a simple issue no doubt but extremely computationally an involved issue. So let us see that how we do the motion estimation, what is the computational complexity that is involved and how we can attempt to reduce the computational complexity without having significant degradation in the prediction performance.

So again go back to the discussion that we had in the earlier class. What we are going to do in the case of motion estimation is that, we must define a search space over which a candidate block has to be searched. We must search for a candidate block and then we must try to find out whether the candidate block matches with that position in the reference block. That means to say that we must have a matching measure. We must have a matching criteria.

Now, what type of motion estimation are we going to employ?
I had said in the last class that we are going to adopt a backward......... I mean, we are going to have a block based motion estimation. So we are going to sub-divide the image into a number of non-overlapping blocks and we are going to estimate the motion for every such non-overlapping block. So it is a block based motion estimation and the matching criterion which are used are as follows.

So the matching criteria for block based motion estimation that is........ one is the mean of squared error or what is called as the MSE mean of squared error then mean of absolute difference this is referred to as MAD mean of absolute difference and then another criterion that is used is matching pixel count or also referred to as the MPC.

(Refer Slide Time: 33:45)

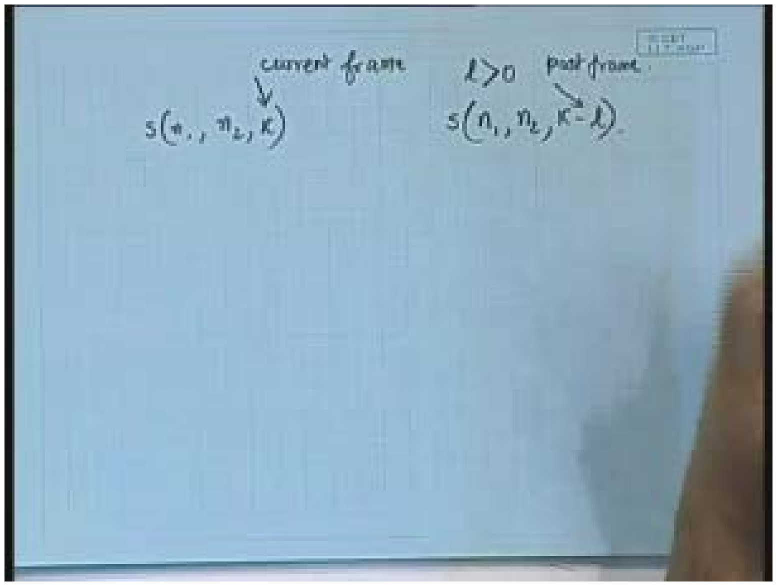


matching criteria for Block-based motion estimation :

- Mean of Squarred Error (MSE)
- Mean of Absolute Difference (MAD)
- Matching pixel count (MPC)

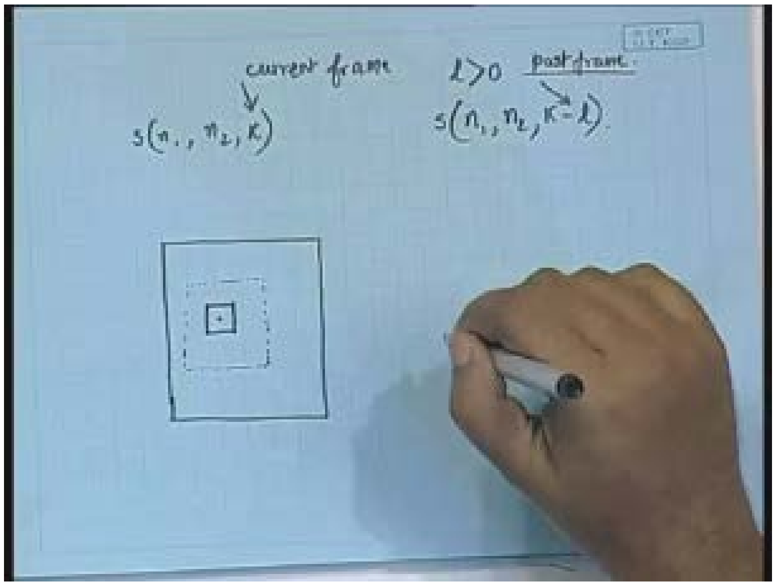
Now, in the MSE or the mean squared error criterion, what we are going to do is that we are using some reference frame. Let us say that the reference frame is a past reference frame. It does not matter whether it is past reference or future reference because we will just change the index accordingly. But what to say is that; supposing we take s (n 1, n 2, k) so k is the current frame. So k is the index for the current frame and for the reference frame we are considering the intensity function as s (n 1, n 2, k minus ℓ) and if we have ℓ as greater than 0 in that case this is a past frame and past frame is used as a reference.

(Refer Slide Time: 34:51)


current frame
$s(n_1, n_2, k)$
$\ell > 0$ past frame
$s(n_1, n_2, k-\ell)$.

Let us say now what we are going to do is something like this that let us say that this is the image and this is a particular candidate block position; this is the coordinate position of a candidate block; let us say that the centroid of this candidate block or the center position of this candidate block is lying over here and we would like to search that in the past frame where this block was so we define a search window.

(Refer Slide Time: 35:35)

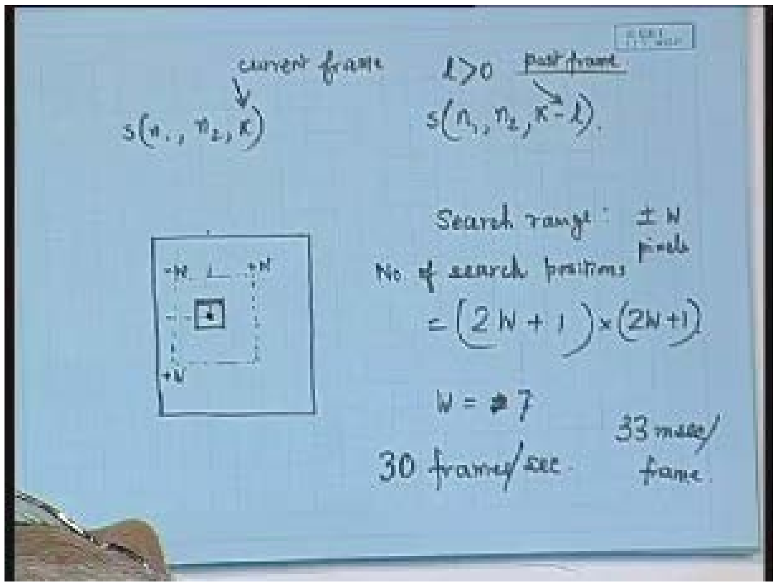


Now, realistically speaking; although one can consider that that block may moved anywhere in the past reference frame but again considering that it is a physical object and the difference between the difference in time between the present frame and the past frame is only a matter of 30 milliseconds or so if we are considering a thirty frames per second rate for the video capture, then every physical object is going to have some restriction about the object motion. So whenever we are considering the search window let us say that this is the search position (Refer Slide Time: 36:23) then let us then there is a particular range over which we should search; there is no point in searching everywhere, so we define a search range and let us say that the search range is plus minus W number of pixels.

Now, what I mean to say by plus minus W is that that if this is the current frame position current block position we should search within a space of this where this goes as minus W to plus W horizontally and again minus W to plus W vertically taking this to be the (0, 0) as the reference so total number of search positions is how many; total number of search positions will be....... yes, 2 W plus 1 by 2 W plus 1 and why plus 1 because this center position is also a position to be searched so 2 W plus 1 into 2 W plus 1 this is the number of search positions number of search positions. this is actually going to decide about our search complexity 2 W plus 1 into 2 W plus 1 so we should have less value for W which means to say that when I have W as plus minus 7 pixels or rather W as 7 so the search range is plus minus 7 pixels then the total number of search positions becomes 2 2 5 225. So 7 is not a large number but 225 search positions is definitely a large number.

Here 7, I mean, going to the left in 7 pixel, going to the right in 7 pixel, going up by 7 pixel, going down by 7 pixel is quite feasible as far as the realistic motion is concerned. But searching for 225 search positions is really difficult. But let us say that we have a very good computer which can compute the motion in real time. Because again mind you; the motion estimation has to be done in real time. You have got only 33 milliseconds at your disposal. Whenever you are having a frame rate of 30 frames per second, you have only 33 milliseconds per frame at your disposal to encode the frame.

(Refer Slide Time: 39:21)



current frame $\quad l > 0 \quad$ past frame

$s(n_1, n_2, k) \qquad s(n_1, n_2, k-l)$

Search range : $\pm W$ pixels

No. of search positions

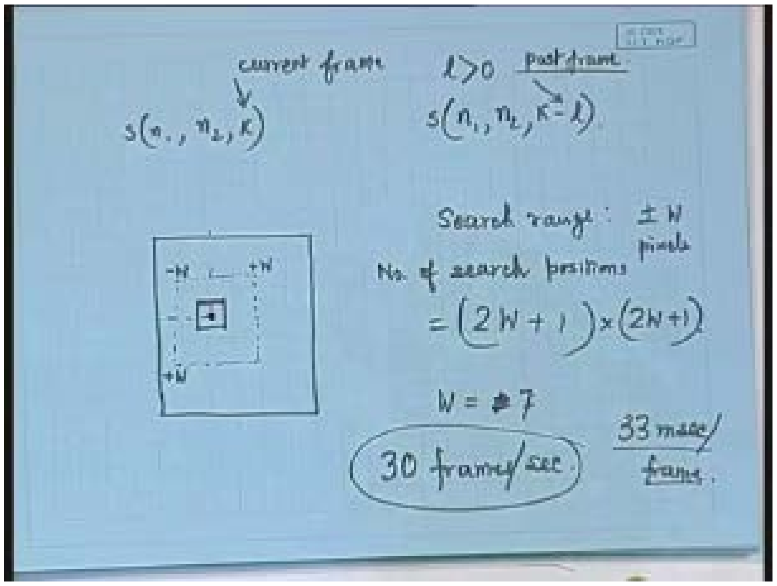
$= (2W+1) \times (2W+1)$

$W = 7$

30 frames/sec. $\quad$ 33 msec/frame.

Now, if you employ a very slow processor or because of having so much of search complexity if you cannot complete your frame encoding within 33 milliseconds of time in that case you are definitely going to lose the real time effect. You have to then skip some frames and you have to lose information, all sorts of problems would arise because of that. So what I mean to say is that let us restrict........ so, for 30 frames per second rate once we are considering that then we should have a real time computation and the real time computation means the computational load should be really considered very carefully.

(Refer Slide Time: 40:13)

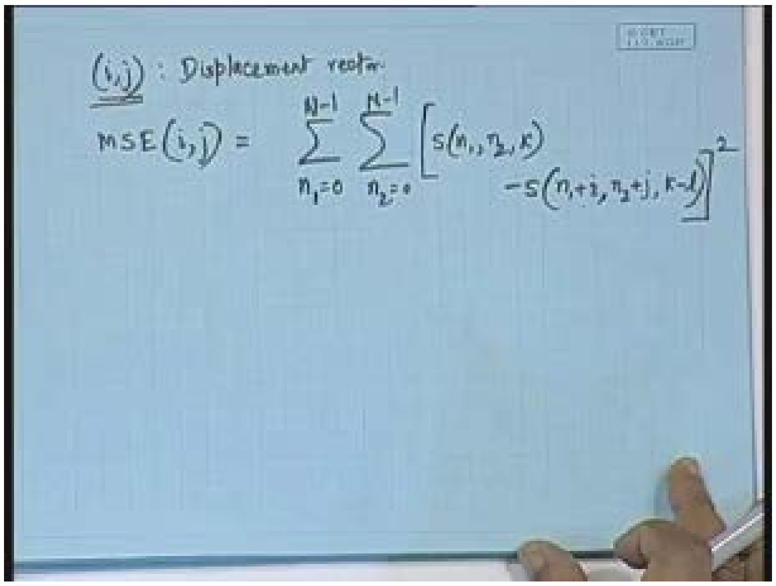


Now what is the computational load?

Now, in order to compute the matching criteria based on the mean square error what we do is that the mean square error measure let us say MSE if we are let us say that we are computing the matching score with a displacement vector of (i, j) so (i, j) is the displacement vector now (i, j) will vary because in the case of the search what we are going to do is that we are first going to place this widow at this position then shift it to the next pixel, then shift it to the next pixel like this we search everywhere that means to say that we have a current displacement that is applied. If I and j both are (0, 0) that means to say that we are searching only at the center position. If I make i is equal to plus 1 and j is equal to 0 this means to say that I have only proceeded in the n 1 direction by one step and in the n 2 direction I have not given any extra displacement.

Therefore, the mean square error at the (i, j) displacement position should be indicated as this that we use the two indices n 1 and n 2 and define it like this: n 1 summation n 1 is equal to 0 to n minus 1 n 2 is equal to 0 to n minus 1 s (n 1, n 2, k) minus s (n 1 plus i, why n 1 plus i because I have taken (i, j) to be the displacement vector so in the reference frame I will go to the position n 1 plus 1 n 2 plus j and the reference frame is the past frame so I consider k minus ℓ with ℓ greater than 0. This is the difference in the intensity at the corresponding position because n 1 n 2 is considered to be displaced at (n 1 plus i, n 2 plus j) so this is the difference in the intensity but we do not deal with the difference in intensity because the difference in intensity could be positive or negative. So if we sum up the difference in intensity over a block of pixels or over the pixels within the block then we may be baffled by a number zero saying that as if to say that there is no error but really speaking it may be just a compensation of the positive errors and negative errors. So errors should be always represented as positive. Now, to do that we do a squaring of this.

(Refer Slide Time: 43:22)



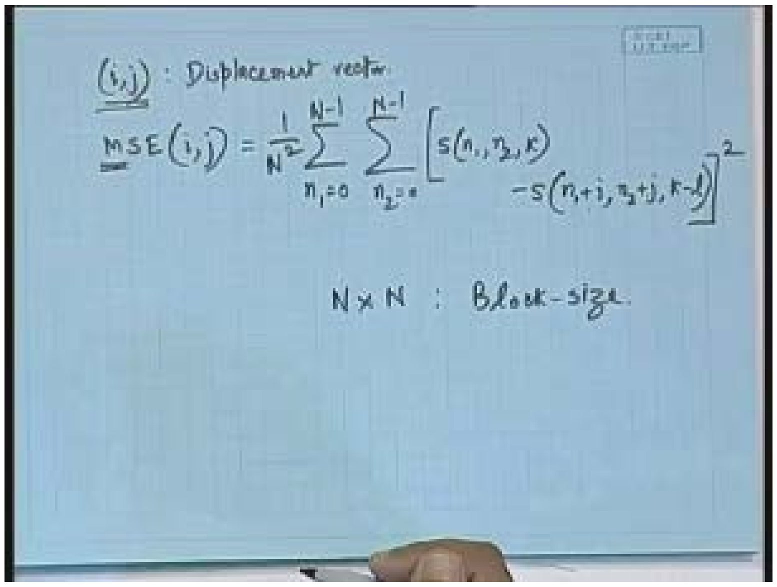
(i,j) : Displacement vector

$$MSE(i,j) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \left[s(n_1,n_2,k) - s(n_1+i, n_2+j, k-\ell) \right]^2$$

But we are saying it as mean square error so what we have done is to sum up this error square or the squared error over the entire widow. So in this case we have assumed N by N capital N by N to be our block size. Again just like the way we used; for the transforms we had used N is equal to 8 similarly one can use N is equal to 8 as the block size; although in the recent standards people have been using variable block sizes so N has come down to 4 or N could be increase to 16 so we can have a search over 4 by 4 as small as 4 by 4 or as large as 16 by 16.

That means to say that if N is equal to 8 in that case this summation computation would involve 64 additions; N square number of additions and this is mean square error so we must divide the whole quantity by 1 upon N square. So the mean square error definition is this.

(Refer Slide Time: 44:36)



(i, j) : Displacement vector

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

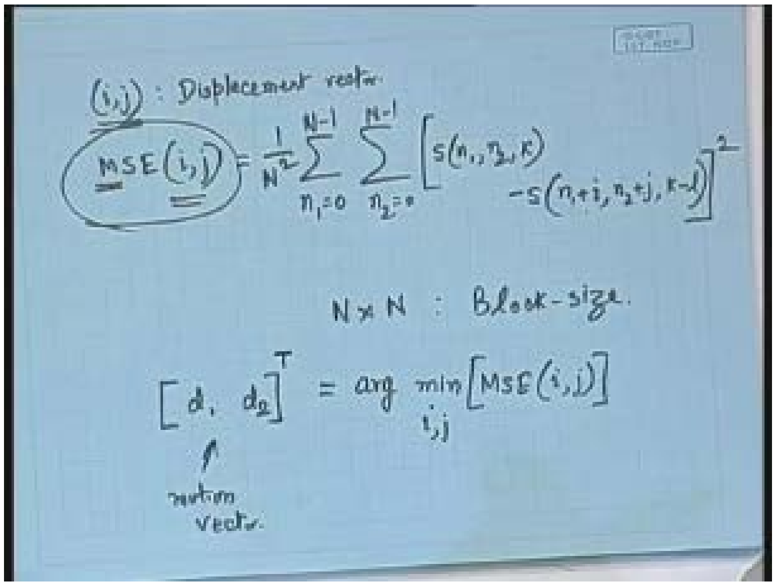
N × N : Block-size

Now this is at a particular displacement vector position (i, j) but I must change this displacement vector while searching I must change this displacement vector so that it spans the entire search window (Refer Slide Time: 44:49) and by exploding several such values of i and j which value I should consider; I should consider that value where the mean square error happens to be the minimum. So to do that what we do is that we consider d 1 and d 2 to be the displacement vector or rather to say d 1 and d 2 to be the motion vector, so this is the estimated motion vector.

And how the motion vector is defined?

We must find out the minimum value of the MSE (i, j). But the minimum mean square error value is not important, what is important to us is the position at which the minimum occurs. Now the (i, j) position for which the minimum occurs that should be my d 1 d 2 value. So when I take this function; when I take the minimum of this function, in that case the minimum must be computed over i and j but i and j happens to be the argument of this mean square error so I should search for the argument that leads to the minimum value. Thus, d 1 d 2 should be defined as the argument minimum over (i, j) MSE (i, j). This is the way by which we can compute the mean square error.

(Refer Slide Time: 46:34)



(i,j) : Displacement vector.

$$MSE(i,j) = \frac{1}{N^2}\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}\left[s(n_1,n_2,k) - s(n_1+i,n_2+j,k-1)\right]^2$$

$N \times N$: Block-size.

$$[d_1, d_2]^T = \arg\min_{i,j}\left[MSE(i,j)\right]$$

motion vector.

Now you can realize one difficulty in this process that we have not only 64 additions but 64 number of squarings are also involved because there is a square term in this MSE computation so 64 squaring are to be done or 64 multiplications and multiplication is of course a very costly resource.

If you are doing multiplication in software you are going to take enormous amount of time; if you are using hardware multiplier, multiplier within the chip and all these things in that case the hardware is going to be very costly. So, in order to have a measure that does not use this does not involve this squaring in that case a very similar idea can be employed but instead of taking the square you can take the absolute difference; absolute difference in the intensity so that also would ensure that the sum total or the individual difference values would be positive and the sum total also will be some positive error. That is actually used in the mean of minimum of absolute difference criteria. So minimum of absolute difference criteria I have mentioned; minimum of absolute difference or mean of absolute difference and taking the minimum of that (Refer Slide Time: 48:09) is the MAD measure and the MAD measure is defined like this that MAD at the position (i, j) would be given as 1 upon n square summation n 1 is equal to 0 to capital N minus 1 n 2 is equal to 0 to capital N minus 1 and in this case I take the mod s (n 1, n 2, k) minus s (n 1, plus i, n 2, plus j, k minus ℓ) mod of this. So this is the MAD criteria.

Matching pixels curve is another measure; I need not have to elaborate the MPC because MAD is something which is most commonly employed in most of the almost all the standard implementations which you will find reported in the literature or even in the standards what they recommend is to compute the MAD. Because, the advantage here is that you are computing the difference but you do not require any multiplications; you do not require any squaring so all that you will need is only the adder and subtractor. Using a subtractor you just take the difference of this value, take the absolute difference of that and then you just add up the values. So this will involve 64 additions and 64 subtractions.

So till this much we learnt in this class and in the next class we will see that how efficiently the matching can be performed and also a better idea about the video coding standards, thank you.