Digital Voice and Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

Indian Institute of Technology, Kharagpur

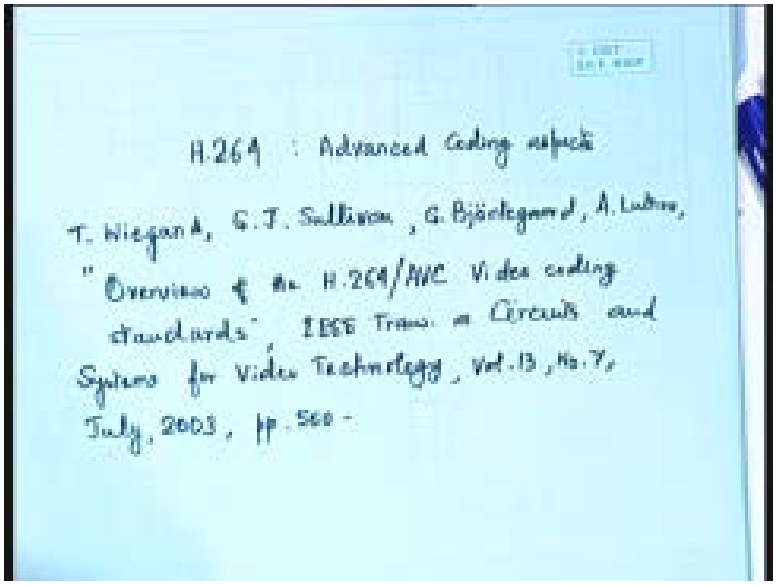
Lecture - 27

H dot 264: Advanced Coding Aspects

..... and I think that will a give you a very important concept of how exactly the H dot 264 is able to achieve the kind of compression figures which we were mentioning that means to say that with respect to its predecessors that's to say the H dot 261 H dot 263 and also with respect to the MPEG-1 and MPEG-2 standards where H dot 264 has clearly got a very distinct advantage in terms of performance. So we are going to have some advanced aspects of it.

Now before we discuss further here are some of the very useful references which you can consult to learn more about the H dot 264 standards. Of course some of the ideas which I have discussed and which I will be discussing in this class are already adopted from these references. But just for you to note down, one reference is by Thomas Wiegand T. Weigand, G. J. Sullivan and G. Bjontegaard and Ajay Luthra; the title of the paper is Overview of the H dot 264 H dot 264 Oblique AVC where AVC stands for the Audio Visual Coding AVC Video coding standard. This gives in considerable details about the different aspects of H dot 264 all the highlighting aspects and this paper appeared in IEEE transactions on Circuits and Systems for Video Technology, Volume 13 number 7, July, 2003.

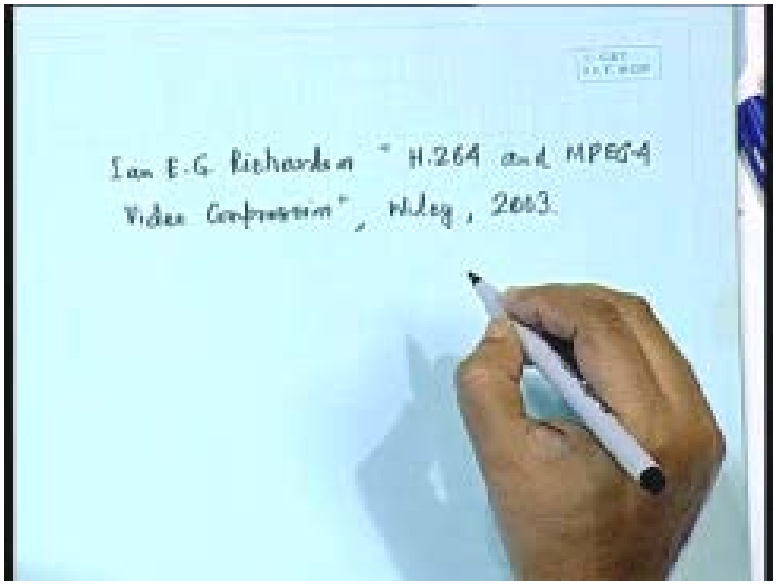
(Refer Slide Time: 4:27)



As a matter of fact July, 2003 happens to be the special issue on the IEEE transactions on Circuits and Systems for Video Technology so those who are interested in knowing various aspects of H dot 264 are very strongly recommended to go through this particular IEEE journal paper very thoroughly because you will be finding lot of useful references there. The page number for this paper that I am talking of that is 560 to 576. This team lead by Thomas Wiegand, they are the makers of this standard so they are very much in the process of designing this standard so naturally a tutorial paper from them is definitely a very good reference.

And another reference that I would like to suggest for you is a very recent book that has been published from Wiley and that is by Ian E. G. Richardson and the title of the book is H dot 264 and MPEG-4. In fact it covers both these standards in very considerable details. In fact I did not cover MPEG-4 much, only just made a very brief mention of it. H dot 264 and MPEG-4 video Compression that's the title of the book by Ian Richardson and this book is published by Wiley so it is an international edition book by John Wiley and this also was published in the year 2003 so Richardson also happens to be a person in the process of this standard.

(Refer Slide Time: 6:07)



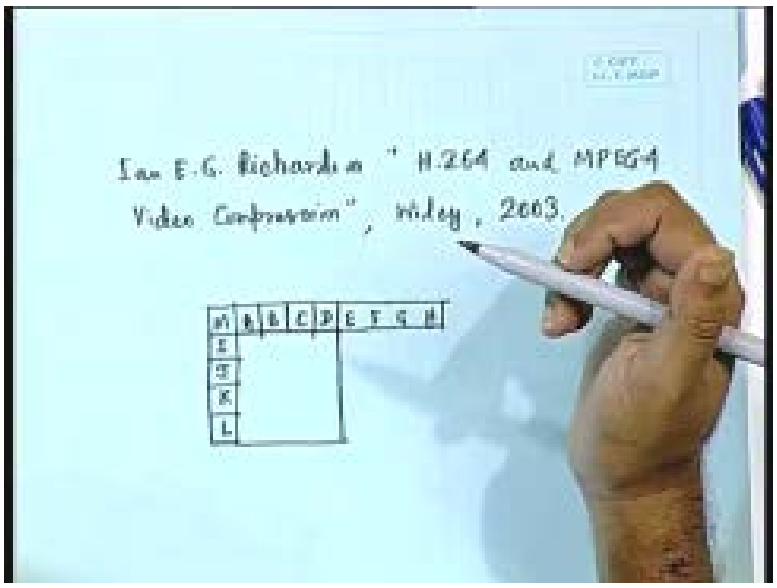
Now what I was mentioning about H dot 264, I mean, where it has advantage over its predecessors are not just a single breakthrough achievement but it is a combination of several different factors that has laid to the high performance.

I have mentioned about the quarter pixel accuracy which is a definite improvement over the half pixel accuracy motion estimation which was done in the past. I made a mention about the variable block size which is a very significant achievement in the sense that depending upon the size of the moving objects one can decide that whether to use a very small block size or whether to agglomerate the small blocks of 4 by 4 into a size as large up to as large as 16 by 16 and also I made a mention that as compared to its predecessors there is an intra-prediction mode and in fact intra-prediction mode happens to be based on the directions so we had specified eight different directions numbered 0 to 8 which covers not only the horizontal and the vertical direction for prediction which means to say that depending upon the direction that you select the coefficients which are immediately on top will be propagated either vertically or horizontally or at an angle of 45 degree or 22 and a ½ degree and I also said that there is a mode 2 which happens to be the DC predictions so some people were having the having some questions that what is going to be this mode 2. In fact mode 2 is a DC prediction so let me make a mention about that.

In mode 2 what is being done is something like this that supposing we have a 4 by 4 block which is to be predicted intra-predicted (Refer Slide Time: 8:32) so we are labeling the pixels are as follows that the previous in the previously received line we are calling this as the M; M is the top left to it then A B C D then we also consider E F G and H, why we also have to consider E F G H that also I will be just telling you so that the concepts are more clear to you and these pixels we are labeling as I J K L. So these are the pixels which are the already received pixels and which is at the boundary of the 4 by 4 block to be encoded.

Mind you, very purposefully we are not dealing with these neighbors or the neighbors below, why, because these are the neighbors which we will be receiving later. I mean, in the raster scan order these are the pixels that have been already obtained and these are the future ones so that is why we do not take that into consideration.

(Refer Slide Time: 9:52)

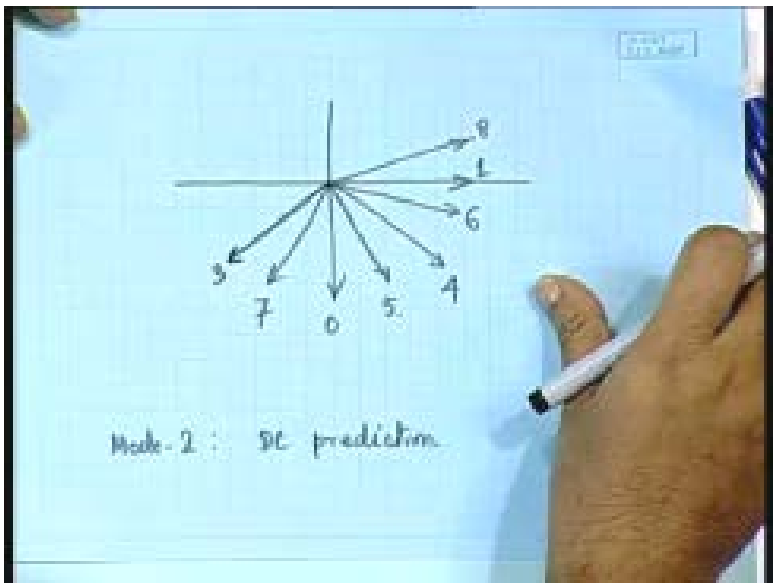


Now in the case of DC prediction what is been done is that the mean of A to D and I to L that is computed so this is what it will be so the entire block the entire 4 by 4 block will have the same value so it is a DC value and what is that DC value that DC value will be the average of these four plus these four that means to say eight pixels, so you add up these eight pixels and add them

up and whatever is the value the predicted values will be like this. So where are you going to use the DC prediction? Definitely when you find that the block which is to be intra-predicted that is pertaining to a very smooth area and there is hardly any variation with respect to its horizontal neighbors or vertical neighbors then it makes much sense in going in for the computation of mean of A to D and I to L in mode 2.

So mode 2 as I was telling you that although it is numbered as 2 it is not exactly a directional prediction but it is a DC prediction and the other mode side I already mention that mode 0 is where we are vertically propagating down; A B C and D will be propagated like this whereas in the case of mode 1 that is the horizontal propagation so all the predicted 4 by 4 values will be generated by propagating the I J K and L along the rows and then diagonal also is very clear meaning that whenever you are having direction 3 direction 3 means diagonal left down diagonal down left, so, as the name implies diagonal down left so the direction-wise you can see that this is going to be the mode 0 and this is going to be mode 1 I can just take out the picture if I.....

(Refer Slide Time: 12:22)

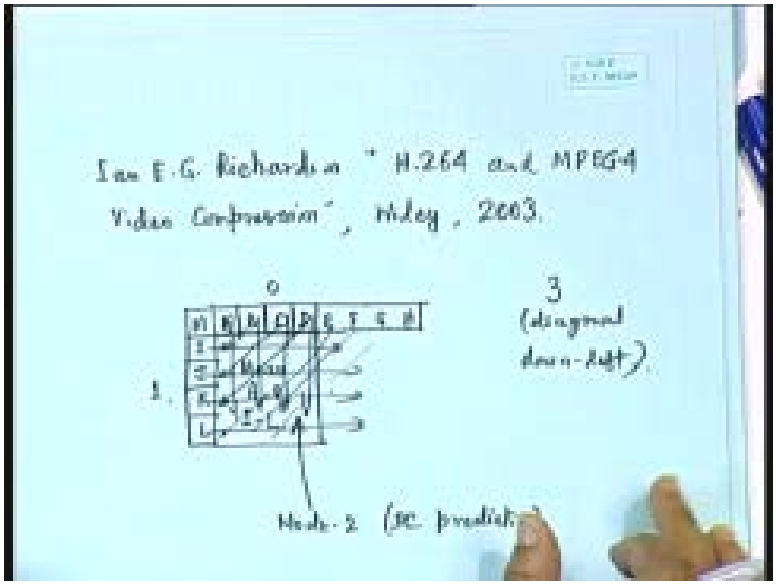


So direction 3 as I have shown direction 3 is this one so it is diagonal down left. So how can we have directional propagation of diagonal down left?

What is been done is that in that case B comes here, C propagates along this, D propagates along this, E propagates along this so this is where the pixel E would come into picture; pixel F and pixel G those will be involved in the prediction in the diagonal left direction and likewise in the diagonal down right direction also this M will get involved along with the A B C and D. This is what one has to consider so this is just to explain why we require the mode 2.

So, depending upon the intensity pattern, like say for example if you were finding that the image is having more of vertical stripes like thing then it makes more sense in doing a vertical propagation that means to say a mode 0 prediction; if you were having predominance of horizontal stripes go in for mode 1 propagation and if you are having some kind of diagonal stripes go in for mode 3 so like this it's to be decided.

(Refer Slide Time: 14:06)



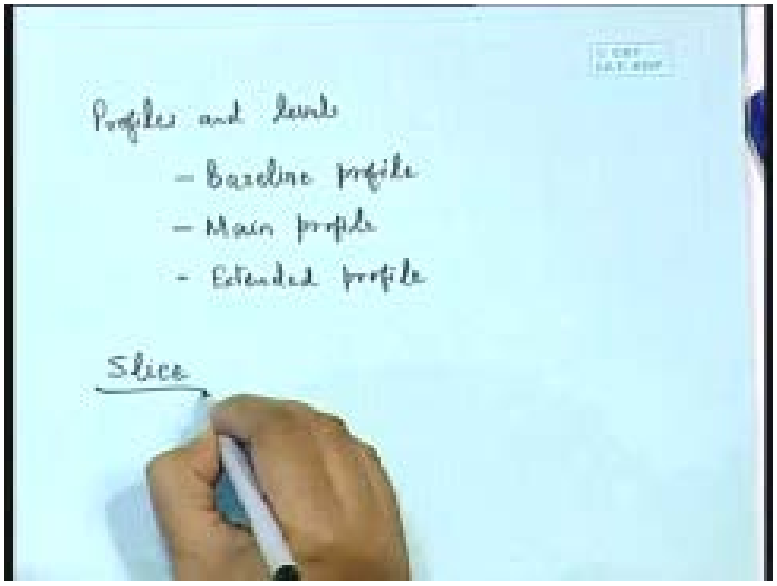
Now we are going to take up some more advanced aspects of H dot 264 and in that we first discuss about the profiles and levels in H dot 264 profiles and levels. Now as is always the case with every standard that the standard specifies different profiles so that depending upon the application one can select either a very simple profile or can go in for more advanced profile. And in fact in the case of H dot 264 it supports three profiles.

The fundamental one is what is referred to as the baseline profile. In fact this is the simplest version of H dot 264.

Now, further to baseline profile means a little more involved and advanced is the main profile and even more advanced is the extended profile. Now in these three different profiles one has to see that which are the features that one includes with the baseline. So naturally baseline is going to include some less features as compared to main profile. So you can obviously expect that the main profile and the extended profile will include whatever the baseline profile is already providing and the extended profile should also include some of the features of the main profile.

Now, in matter of profile, I mean, before we discuss that how this baseline profile, main profile and extended profile characteristics are formed before that it's very important on our part to introduce the concept of slices.

(Refer Slide Time: 16:30)

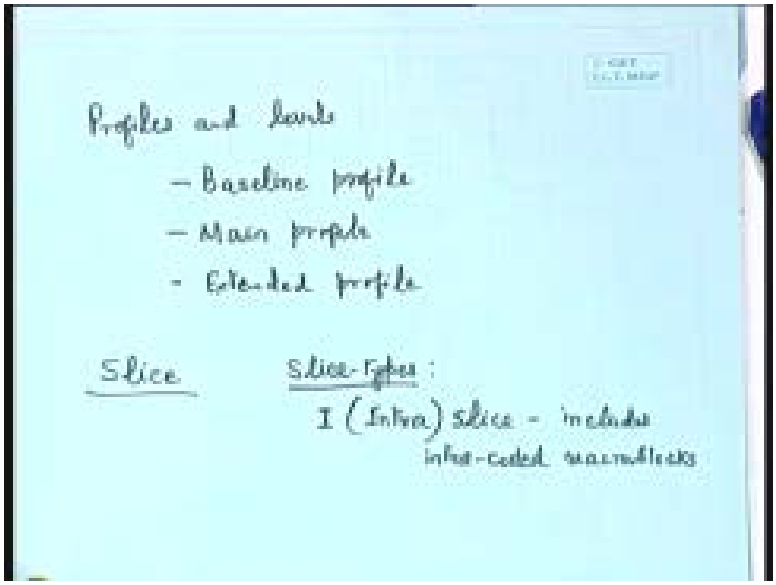


Now I think that I made a very brief mention about the concept of slice in one of my earlier lectures and there I had said that slice is nothing but a group of macro blocks macro blocks and that may be or may not be a combination of macro blocks in the raster scan order; means you

may have the macro blocks in exactly the raster scan order but in the case of H dot 264 that is not mandatory, in fact you can have the slices which can be grouped and which can be grouped into what is called as the slice group and the slice group need not include the slices which are in the raster scan order.

I will give you some examples of this. Let us first talk that what are the different slice types that is supported by H dot 264 and the slice types that it supports are like this. The first is the I slice which is referred to as the intra slice. Now, intra slice basically includes the slices includes the macro blocks which are intra-coded so I slice includes intra-coded macro blocks.

(Refer Slide Time: 18:16)



In fact the whole idea of..... I mean, you may ask that why is this grouping necessary why are we or what big purpose it is serving by grouping the macro blocks into slices. Well, what happens is that for a slice we can define a data structure and a major advantage of using the slice is that we can give unequal importance to the different slices like slices which correspond to the macro blocks in the foreground area that is more important to us in the bit-stream as compared to the slices which are not in the foreground area. So, as a result of that one can introduce what is called as the unequal error protection and one can decide that one can tolerate some amount of

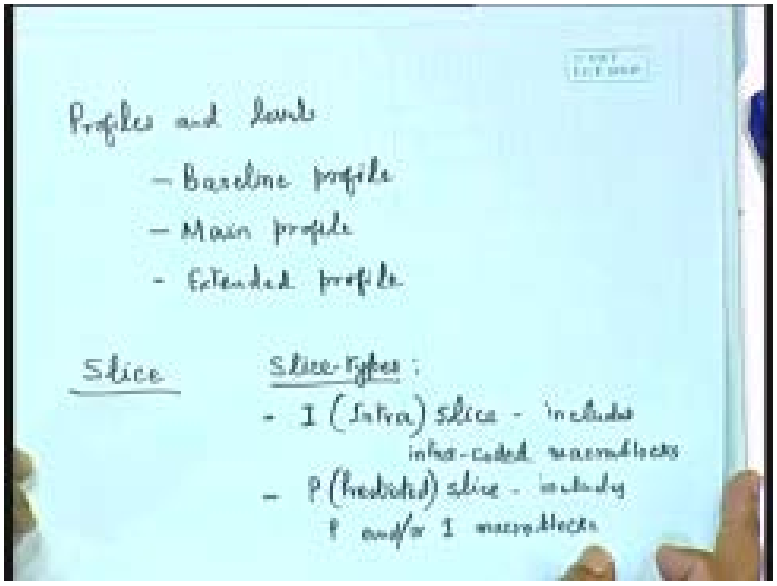
redundancy while transmitting the foreground macro blocks or the foreground slices as compared to the background slice where the quality may suffer; we need not have to give that much of redundancy for the error protection because even if there is..... in one or two frames even if there is error in the background that may not be very much noticeable.

So, to every slice group we are ultimately going to attach the bit-stream and that bit-stream will be encapsulated with some error protection bits before it is actually put into the bit-stream. Again one aspect which I need to talk at this point is that how many macro blocks should be there in one slice?
Well, there is ample of flexibility. You can have a slice define with a single macro block; a slice may include one macro block or more than one macro block and then also how many; I mean, when we say more than one what is the maximum limit? Well, the maximum limit is the full image size means for the entire image you can define one single slice.

See, the idea is that if you transmit more number of slices you are getting the advantage of incorporating unequal error protection and you can attach importance to the different slices that is an advantage but in the process you also incur some amount of overhead in the sense that with every slice transmission you have to give some header information so you will be definitely incurring those header information as an extra overhead. So, from that point of view if you are able to reduce the number of slices then it is definitely better.

One of the slice type is the intra or the I slice, the other is the P slice or this is as the name implies it is a predicted slice and the P slice not only includes the predicted or the P macro blocks so it includes P and oblique r I macro blocks so P slice can include the I macro blocks also in addition to P. this is about the P slice.

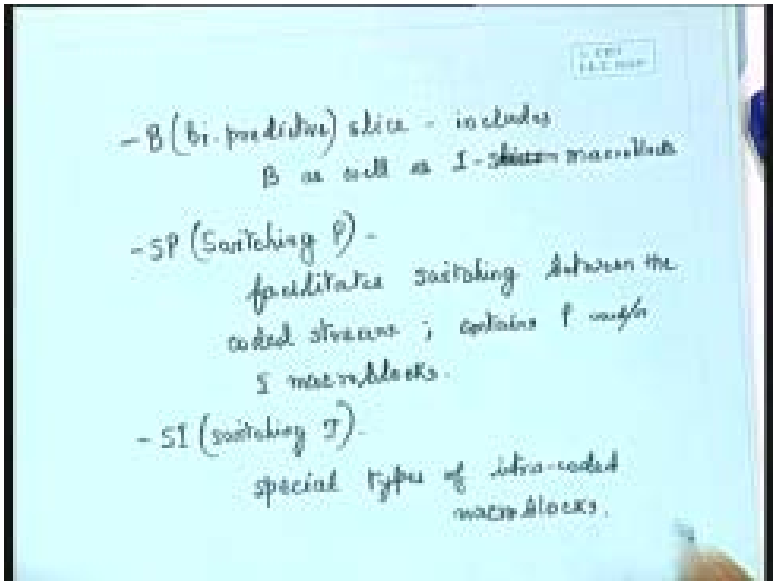
(Refer Slide Time: 22:08)



There is also the B slice and B refers to the bi-predictive bi predictive slice and the B slice that includes that includes I or that includes B as well as I slices. In fact as I was mentioning that H dot 264 has got a provision for multiple frame reference, now B slice obviously requires two different references: one is the past reference and the other is the so-called future reference; of course not exactly future anything because we are introducing a delay so that we have so with respect to the frame being encoded which is already an old frame we can take one of the reference as a past frame and one of the reference as a frame which is occurring later than the frame being encoded that is the by predictive or B mode of prediction we use.

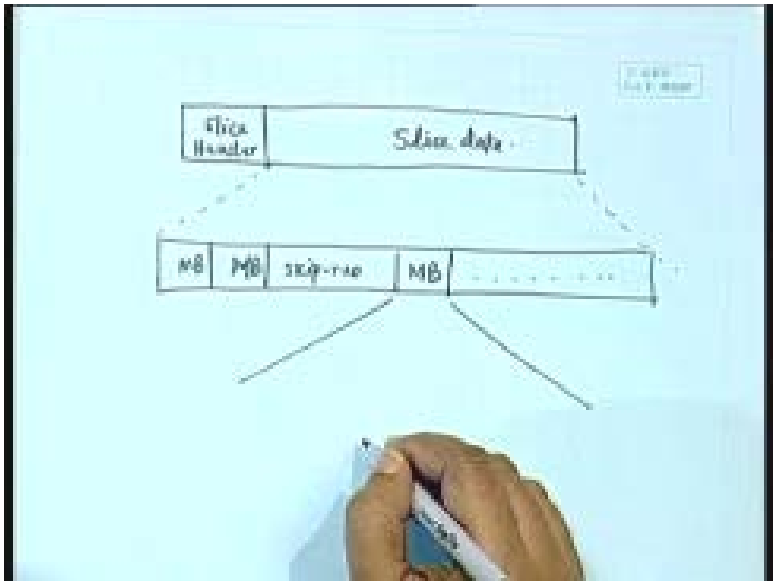
Now in B slice we include the B as well as I macro blocks sorry it should include B as well as I macro blocks and then also there are two very advanced slice types: one is what is referred to as the SP whose full form is the Switching P slice; this facilitates switching between the coded frames facilitates switching between the coded streams and in fact the coded streams can contain P so you may have a coded stream composed of the P macro blocks, you may have a coded stream consisting of I macro blocks and these two coded streams can be switched in the SP or the switching P mode and there is yet another mode which is called as the SI or the switching I and in fact it is consisting of a special type of intra-coded macro blocks.

(Refer Slide Time: 25:31)



Now these are the I, P, B, SP, SI these five are the different types of slices which is supported under H dot 264. And the slice syntax that goes something like this that for every slice we have a slice header at the beginning so this is the slice header (Refer Slide Time: 26:04) and the slice header is followed by the slice data and the slice data that consists of the macro block information because slice as I told you already that slice is composed of macro blocks so there will be the macro block information and not all macro blocks are going to be encoded because as I was telling you that there is also what is called as the skipped macro block mode so there will be some skipped run off macro blocks and then again there will be some macro blocks which are to be encoded and this is how it goes.

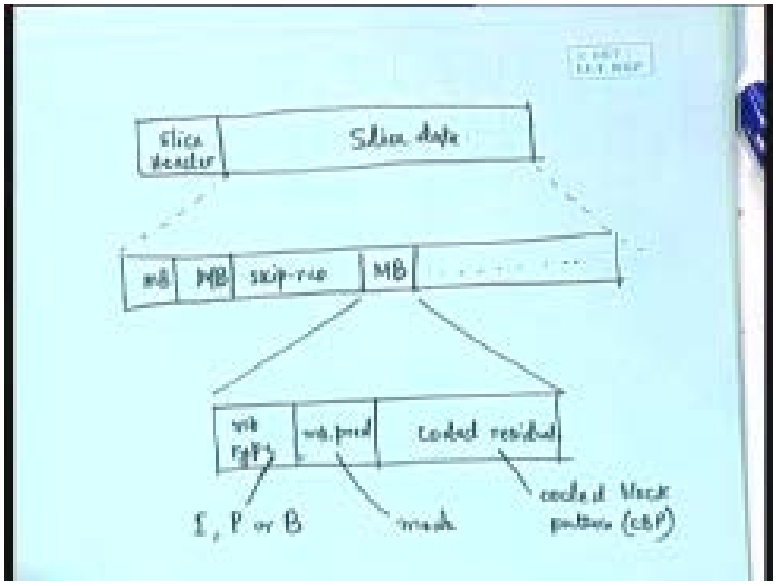
(Refer Slide Time: 26:59)



Whenever you see the data structure of the macro block or the macro block syntax when you find then you will find the presence of these fields. First of all you have to specify the macro block type. So I write here mb type. So basically this field indicates that whether the macro blocks is coded in intra or inter and again in inter it could be P or B. So whether it is a I macro block, P macro block or B macro block that will be written in the mb type information then it also has a field for the macro block prediction mode. So mb pred basically specifies the mode of prediction and I have already talked about the mode of prediction both in the case of intra as well as inter-predicted macro block.

Intra-prediction of macro block I already mentioned about the directions and the DC that means to say all those nine prediction modes for intra and also for the inter-predicted macro block we have the variable block size which I have already mentioned for the inter-prediction and then there is a then we have to transmit the residual information so this is the coded residual. in fact it consists of the coded coefficients and the coded coefficients are in fact represented in the form of what is called as the coded block pattern or in short form it is referred to as the CBP this is what forms the hierarchical data structure of the H dot 264 data.

(Refer Slide Time: 29:20)



So slice is one layer above the macro block, within macro block one has got a very specific data structure so this is basically containing the coded block pattern of the individual blocks that composes the macro block.

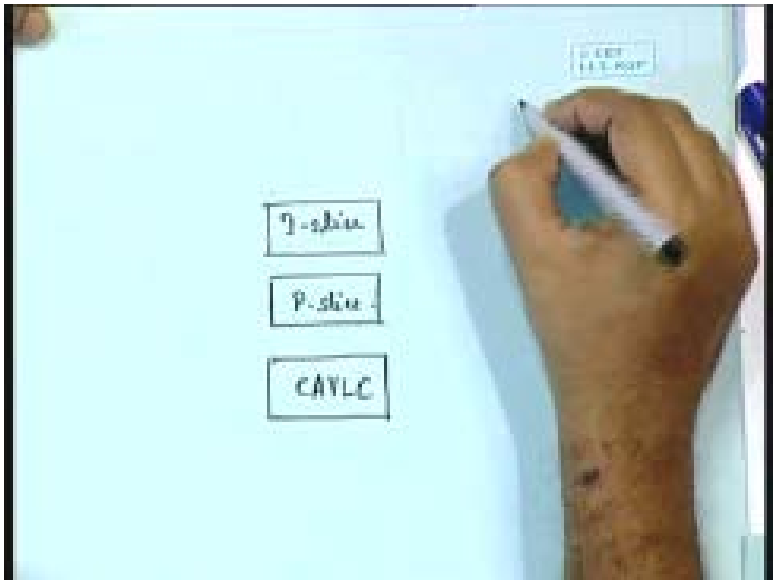
Now how many blocks again that is also flexible because macro block is a 16 by 16 luminance, 8 by 8 chrominance from the two chrominance channels whereas 16 by 16 itself may contain up to sixteen luminance blocks or it could be less than that depending upon how we partition the macro block; whether we have a single 16 by 16 or whether we have two 16 by 8 or two 8 by 16 or four 8 by 8 and every 8 by 8 again can be partitioned as I already mentioned.

Now coming back to the question of the profiles that are supported under H dot 264 the baseline profile basically supports the I slice, the P slice and also a very important aspect which I wanted to discuss slightly that is about the entropy coding aspect of H dot 264.

Now, every standard every video coding standard has got two parts: one is the lossy coding part where the transformation is applied and transformation is followed by the quantization and all the quantized coefficients are then sent in the form of lossless encoding so there we adopt and

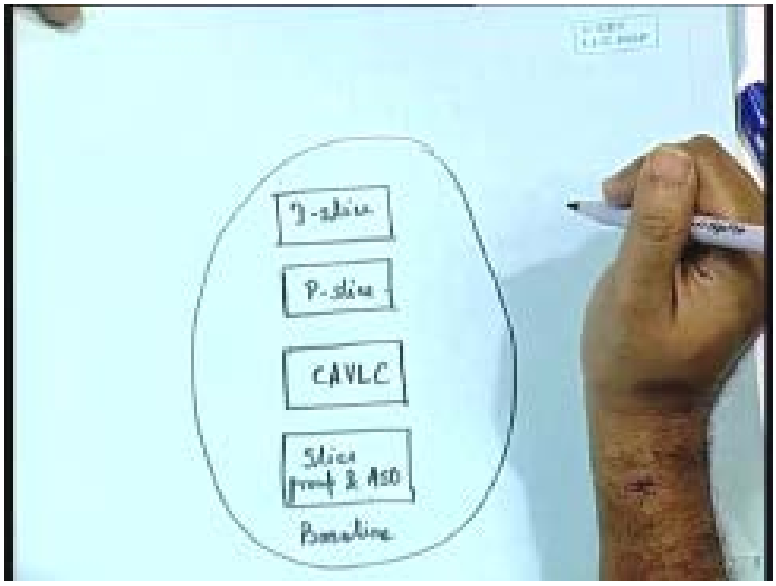
entropy coding scheme and H dot 264 also has got the entropy coding scheme but it is advanced with respect to what the earlier standards have supported. Earlier standards have talked about the Hoffman coding or the arithmetic coding and H dot 264 has gone a step further; they have proposed what is called as the context adoptive variable length coding, in short form it is referred to as CAVLC.

(Refer Slide Time: 31:59)



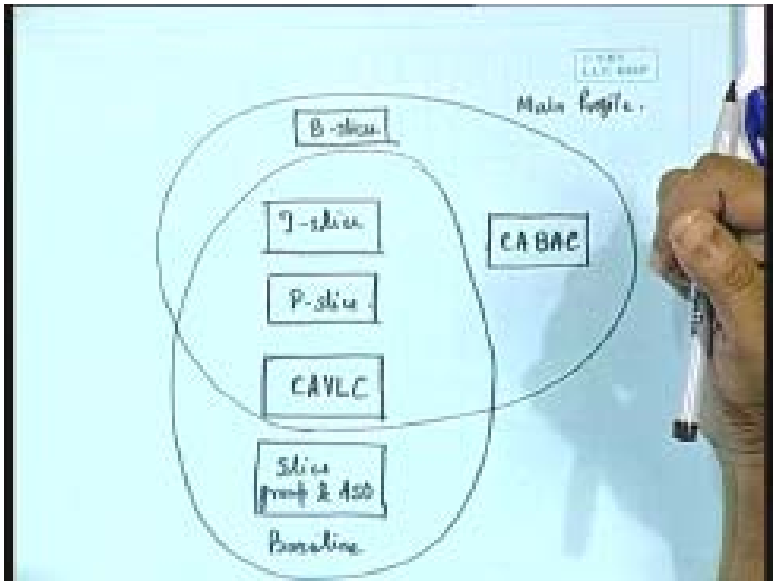
Hence, CAVLC's full form you remember, I am just writing down; CAVLC stands for the Context adaptive variable length coding; variable length coding you already know because all of this that means to say the arithmetic coding or the Hoffman coding they are all variable length coding, any entropy coding scheme has to talk of the variable length coding and then also there is the baseline also supports what is called as the slice group concept, I will just come to that slice group and what is called as the Arbitrary Slice Ordering ASO and these basically compose the baseline profile.

(Refer Slide Time: 33:00)



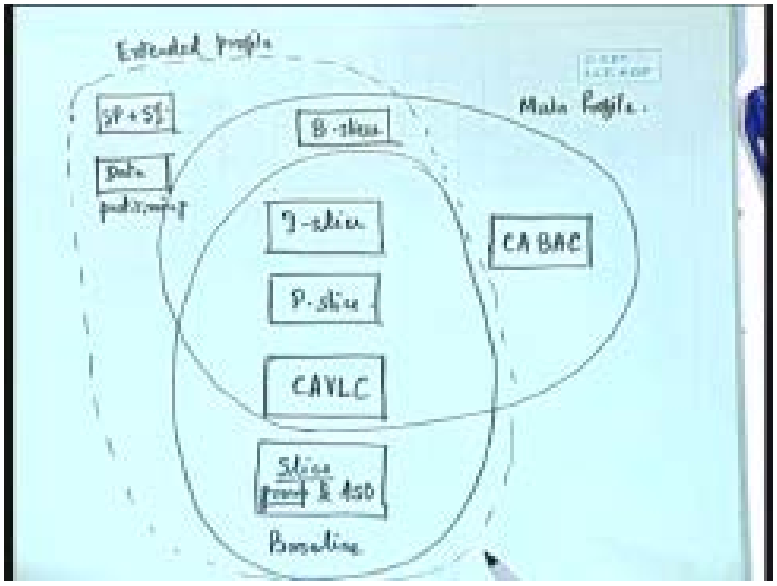
So you can find one omission that in the baseline profile we do not have the B slices, we only have the I slice and the P slice not the B slice, in the entropy coding it adopts CAVLC. So this is in the baseline whereas in the main profile the B slices are included. In fact B slice will not contain this slice group aspect but the main profile..... Now I am showing the components of the major components of the main profile (Refer Slide Time: 33:33). So in addition to I slice and P slice it also contains it also supports the B slices. Then in addition to CAVLC which is already covered in the baseline profile as well as it is included in the main profile we have got another entropy coding scheme which is referred to as CABAC C A B A C and the full form of C A B A C is Context Based Arithmetic Coding Context Based Arithmetic Coding. This is also an entropy coding scheme. The main profile supports both CABAC as well as CAVLC and mind you, you can see some omissions; in the main profile also we did not talk about the SP and the SI slices which I mentioned little while ago and the data partitioning.

(Refer Slide Time: 34:53)



Now why is data partitioning important; again see, the bit-stream may be divided with more than one number of partitions. And as I was telling you that because of this nature of data organization it is possible that you can send you can transmit different slice groups into different channels; they could be entirely different physical channels altogether which requires data partitioning that where exactly you partition the two data and what should be the header information corresponding to every participation. So that is in fact supported under the extended profile. In fact the extended profile supports only the CAVLC as the entropy coding and in the extended profile as I was mentioning that these two slice modes are supported that means to say the SP and the SI these slice modes are supported and also the data partitioning, it does not include CABAC. This is the data partitioning.

(Refer Slide Time: 36:30)



So these are the three major profiles or three profiles in fact which are supported by H dot 264.

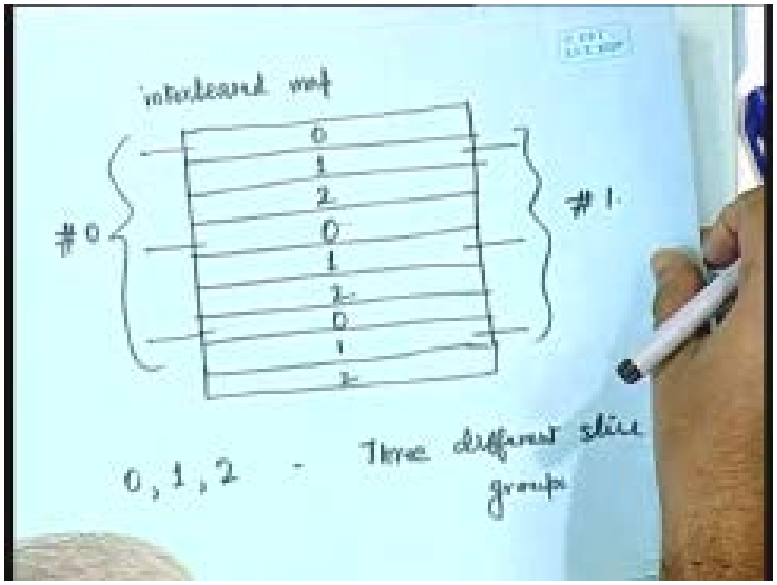
Now I need to talk about slice group; I have mentioned about that but I have not talked about it. So for now slice group as the name implies is actually a collection of slices. Slice in turn is a collection of macro blocks and slice groups are collection of the slices which may not be in the raster scan order means I can give you some examples of slice groups. May be rather than explaining in words a few examples of how to form the slice groups would make it very clear.

First let me give a situation like this that say this is a video frame which is to be encoded and let us define three different slice groups, so we label them as the slice group 0, slice group 1 and slice group 2 so there are three different slice groups that we take into consideration and we can partition this frame into a number of slices. So we can now say that this particular slice (Refer Slide Time: 38:14) the top most one that belongs to the slice group 0, the next slice belongs to group number 1, the next belongs to group number 2 and again the fourth slice belongs to group number 0, group number 1, group number 2, 0, 1, 2 so what it means to say is that if I now talk of the slice group what am I referring to slice group 0. If I say that I want to transmit slice group 0 what do I mean? That means to say that I am taking this slice (Refer Slide Time: 38:48), I am

taking this slice, I am taking this slice these together these three together forms the slice group number 0.

Again if I take this one, this one and this one then together it forms slice group number 1 and likewise if I take these two, these two and these two then it forms slice group number 2 so what you find here is that the slice groups they form a kind of interleaved map. This is an example where I show that the slice group forms interleaved map.

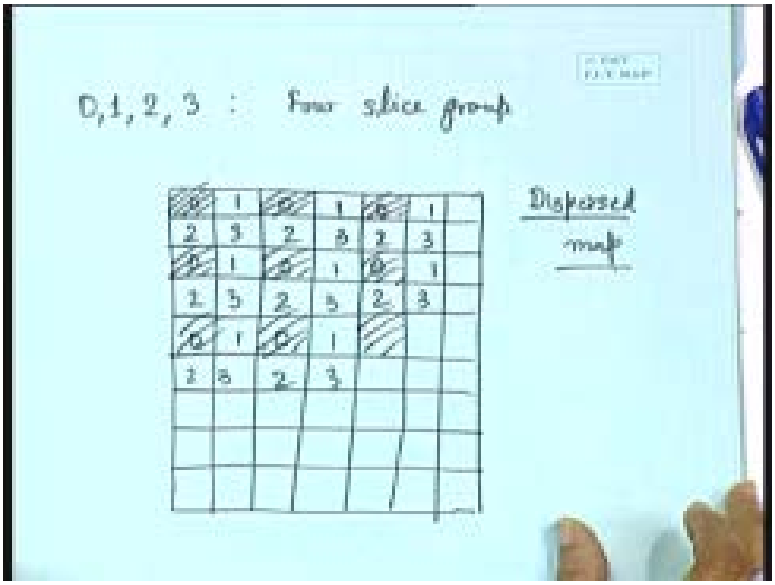
(Refer Slide Time: 39:27)



Now instead of forming an interleaved map like this the slice groups may also form what is called as the dispersed map. So just to show you an example of the dispersed map let us say that we have got four different slice groups numbered as 0, 1, 2 and 3 so let us consider four slice groups. Again how to organize these four slice groups? We can take a frame and slice need not be formed for the entire row of macro blocks; in this case we have formed entire row of macro blocks but we can as well form slice groups like this say we make some horizontal grouping of macro blocks and also we make some vertical grouping of macro blocks; these are not the pixel groupings these are the macro block groupings.

So say this group of macro blocks forms slice number 0 (Refer Slide Time: 40:41), this group forms slice number 1, this group forms slice number 2, this forms slice number 3; again this forms 0 1 2 3; 0 1 2 3 and here I have 0 1 2 3; 0 1 2 3; 0 1 2 3; 0 1 2 3; 0 1 2 3 like that. So now when I talk about slice group number 0 what I am referring to; I refer to this slice, this slice, this slice so all those which are labeled as zeros will be picked up. If I refer to 1 then all those which are labeled as 1 will be picked up so this is actually what is called as the dispersed map so this is a dispersed map of slice groups. The earlier example was an interleaved map of slice groups and this is a dispersed map of slice groups and there could be a realization of slice group in a different way.

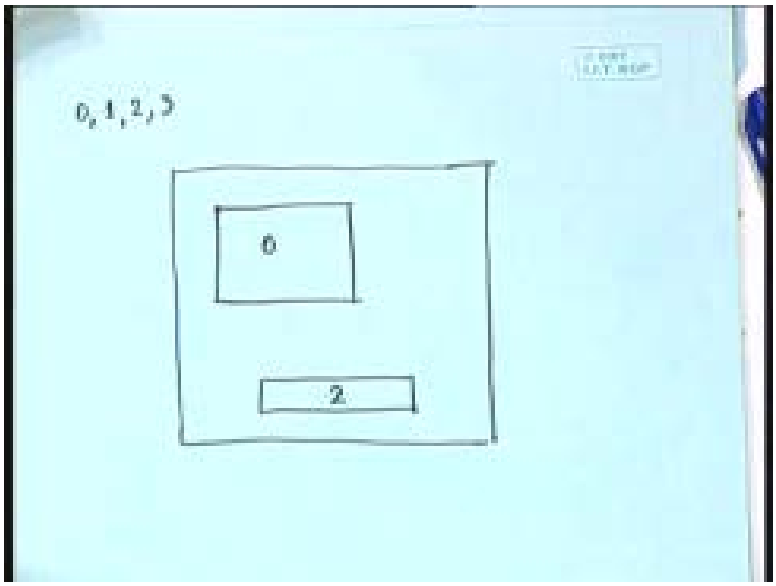
(Refer Slide Time: 41:56)



As I was telling you that slices are very effective for encoding the foreground and the background information separately so you can define the slice groups as something like this. Again take four different slice groups let us number them as 0 to 3. Supposing this is the frame that we have to encode and we can say that this set of slices they form slice group number 0 (Refer Slide Time: 42:38). In order to specify the slice group boundary one can specify the top left the coordinate of the top left macro block and the coordinate of the bottom right macro block; if you if you specify these two coordinates then automatically this rectangular area is

defined and then you can have the slice group number 1, slice group number 2, slice group number 3.

(Refer Slide Time: 43:12)



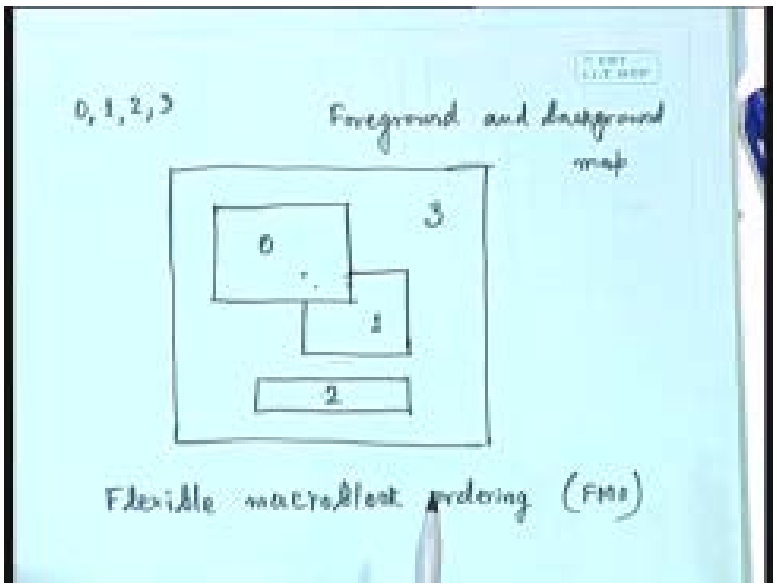
Now supposing I say that these groups of macro blocks they form slice number 2 and for slice number 1's definition I make it slightly interesting. What I do is that while specifying the slice group number one the top left coordinate of the macro block I specify here and the bottom right coordinate I specify over here (Refer Slide Time: 43:30) which means to say that it is going to include all the macro blocks which are falling within this boundary but what about these 1's. These 1's also will come under the group 1 but it is already defined as the group 0 of the slices and in fact lower slice group number has got the overriding precedence; means if I say like this in that case these macro blocks will be under group number 0 and the rest of the macro block will be group number 1, this will be group number 2 and all the remaining macro blocks of course they are not in the raster scan order any more, they also will be labeled as macro block slice as the slice group 3.

Now slice group 3 as you can very clearly see that slice group 3 consists of all the leftover macro blocks. So the leftover macro blocks forms a slice. Now which macro block to be put into which

slice that has to be specified in the form of a mapping and that mapping is what is called as the flexible macro block ordering flexible macro block ordering and in the draft standard it is referred to as the FMO flexible macro block ordering; the final standard refers to the flexible macro block ordering as the arbitrary slice ordering ASO but the basic concepts remain the same.

I think these three different examples make it clear that how one can form the slice groups. This slice group formation is a typical case of foreground and background map. So this is foreground and background map so as you can very clearly see from this example that here we have got three foreground objects; this sort of a situation would be very much ideal for encoding a frame which has got three different objects; one object is occluding the other; the object corresponding to the area of 0 occludes the area number 1 and the remaining region forms the background and we can have the slice group formation based on this.

(Refer Slide Time: 46:16)



Now another advanced aspect that I need to talk about the H dot 264 is what is referred to as the deblocking filter. Now this is a very important aspect so I spend five minutes to just cover up that. This is pertaining to the deblocking filters. You see, one of the major short comings of the H dot 264 or any block based coding in fact is that whenever you are partitioning the blocks into

a partitioning the frame into a number of non-overlapping blocks you have got block boundaries and across the block boundaries you are having some form of discontinuous information which is present in the form of what is referred to as the blocking artifacts. And as I mentioned I believe sometimes back that the blocking artifacts are very much more pronounced in the case of the very low bit rate transmission because whenever we are having very low bit rate transmission and we are compromising on the quality of the coded blocks there is invariably some artifacts which is coming in at the block boundaries.

In this case the number of blocks are becoming more because the block size for the transform coding that we apply the block size is going to be 4 by 4. Now 4 by 4 is very good as far as capturing the local information is concerned because you can have so the preserving the local information within a small area of 4 by 4 is definitely helpful in encoding the details of the image because you are not going in for a codes or blocks size so definitely the details of the image can be encoded better. But also 4 by 4 leads to more number of blocks more number of block boundaries and more number of block boundaries definitely lead to more amount of blocking artifacts.

So, blocking artifact is a serious problem in every standard and more so in H dot 264 because of the reduction in the block size. Now how to take care of that? For this the H dot 264 standard has specified what is called as a deblocking filter and in fact deblocking filter is there in the encoding loop itself and that is why it is also referred to as what is known as the in the loop; in the loop means within the encoder loop; what I talk of encoder loop I think you understand; few classes back when I was explaining about the encoder I have shown you that the encoder has got a forward path for the propagation of the residual coefficient and also it has got a feedback path in order to derive ultimately the predicted signal. So whenever you are making that derivation that time in the loop you incorporate a deblocking filter.

And the basic idea of the deblocking filter is that, if you find that the discontinuity observed in the intensity values across the block boundary correspond to the actual picture detail's discontinuity; if the discontinuity arises because of the picture then you should preserve that detail. But if you find that the discontinuity in the intensity value is not resulting out of the

picture, in that case we should apply a smooth filtering process just to smoothen out the discontinuity.

Now the million dollar question is that how to decide that whether the change in the intensity values are arising out of the artificially created blocking artifacts or whether it is because of the content itself. Well, there it has been observed that whenever there is an there is a discontinuity in the intensity value resulting out of the intensity variations in an actual image there the strength of the ages happens to be much higher whereas in the case of blocking artifacts the strength of the ages are relatively much smaller. So, if we find that the difference in the intensity or rather the extent of discontinuity is less then we say that it is because of the blocking artifacts so we can then apply a smoothening filter and if we find that the difference in the intensity is high then it must be resulting out of the image content and there we do not apply the deblocking filter.

These are some of the advanced features and I suggest that to know more about the details of H dot 264 standards you refer to the references that I have already talked of. This is all about the video coding part that we wanted to talk of so we will be beginning a new chapter from the next class. Thank you.