Advanced Digital Signal Processing - Wavelets and Multirate
Prof. V. M. Gadre
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture No # 38
Two Applications Data Mining Face Recognition

A warm welcome to the 38 th lecture on the subject of Wavelets and Multirate Digital Signal Processing. As promised in the previous lecture, we shall use this session to discuss two applications of wavelets and time frequency methods in great depth. In fact, the two students, based on whose application, assignments, this lecture is constructed are going to discuss, what they only introduced very briefly in the previous lecture. They had kind of given a trailer to their presentations in the previous session, in which they had just explained the essence of the application that they had done.

In this lecture, they shall be explaining the details of the application, and also pointing to some of the results that they have obtained. I must mentioned that these are two students, who have actually use these two lectures to learn the subject, use the lectures over the semester to learn the subject, and have undertaken to explore two applications of wavelets all over the semester, out of a batch of about 15 to 18 students.

The assignments that are going to be presented today have been found to be some of the best in the class. And therefore, in a sense it is also an appreciation of the excellent work that these two groups have done; that they have been invited to record their presentations today. In a broader sense, this is also to encourage, whenever people use these lectures outside, but students should be involved in exploring applications. And we hope that the hard work and the very intelligent efforts putting by these two students; these two groups actually of students, would inspire many other students, who listen to this lecture to explore several other applications of wavelets many as they are.

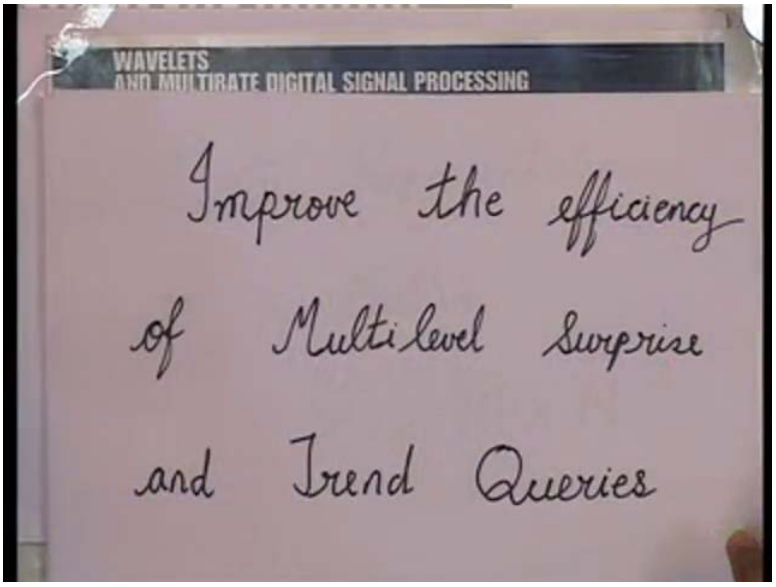
Anyway with that little introduction, let me put before you the two applications that are going to presented in depth today, you have had a trailer of them, but I would like to put them in the broader perspective of the subject. So, in the lectures today, we are first going to look at the application on data mining (Refer Slide Time: 03:04) Kunal Kunal Shah is going to present the application on data mining on behalf of his group of two students, namely he himself, Kunal Shah and Arko Choudhury.

Now, data mining is a generalization of representation. So, in data mining, Kunal is going to show us how one can use the properties of wavelets in efficient representation to advantage in retrieval another such applications from a database. The second application which is going to be discussed today is face recognition (Refer Slide Time: 03:53); now you know there are also going to be certain differences between detection and recognition and so on and the issues; I do not want to take away the thunder as I said from the person making the presentation.

But face recognition is an important, an increasingly important application in many security systems and other image processing or vision systems. So, both of these applications are of great importance in the modern world and we shall now, without much ado invite these two speakers; young student speakers to present the work they done over the semester and to put before you both the concept and the result of what they have done.

I shall first invite Kunal Shah to make the presentation based on the work done by his group of two students, Kunal Shah and Arka Choudhury, thank you. Hello friends, today I would like to talk on the various applications of wavelets in data mining in data mining. As sir said, data mining is used for efficient representation of data, so today I would like to speak on that.

(Refer Slide Time: 05:20)



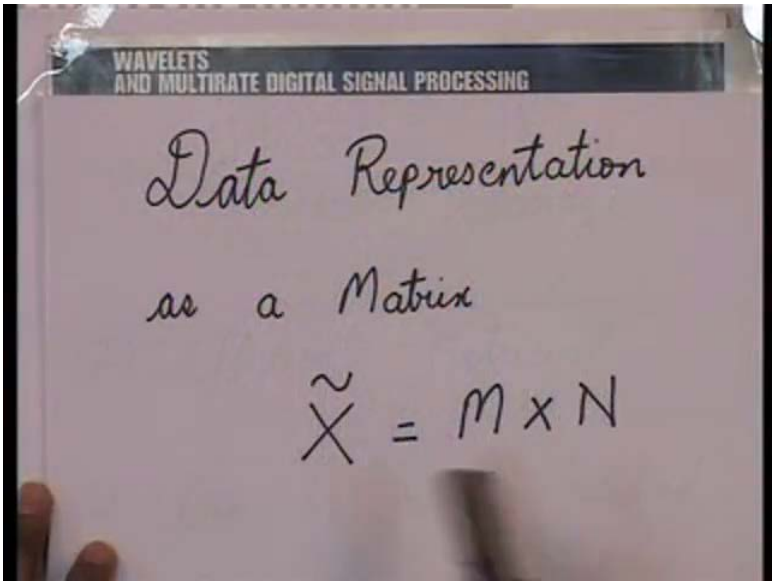
So, the problem statement is given a time series data, the time series data could be huge; our aim is to improve the efficiency of multi level surprise and trend queries from the time series data.

Now, first of all, what is the meaning of surprise and trend queries? Now, when we have a long time series data, generally we do not encounter point queries for example, if we record the temperature of a particular city for a year say, we never ask, what was the temperature at this date of this month. We always ask the trend, how was the change in the temperature during the month, such queries are called as trend queries.

One more type of query which we encounter is surprise query, was there any sudden change in the temperature in a particular city of this month? So, such type of queries we have to handle and wavelets, efficiently handles such type of queries that we have to look.

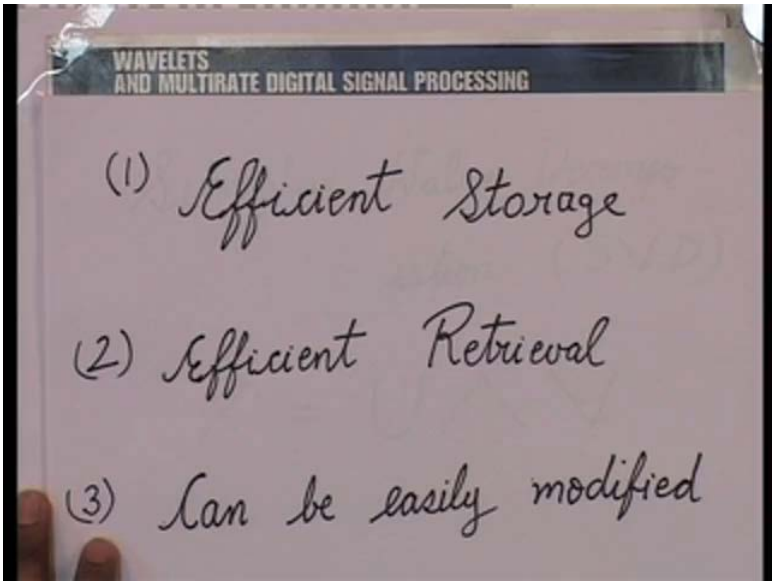
Now, what is the meaning of multi level? Yes, such queries are generally encountered at various level of abstractions, it could be a month, it could be a year, it could be a decade, say if someone asks, what is the change in the temperature during a particular month, we should be able to answer that easily. And if we ask, what is the particular temperature during a decade, sudden change or average in a decade that also we should answer. So, decade, year, month, shows various levels of abstractions. So, we have to improve the efficiency of such type of queries, so first of all, how such huge data is represented.

(Refer Slide Time: 07:15)



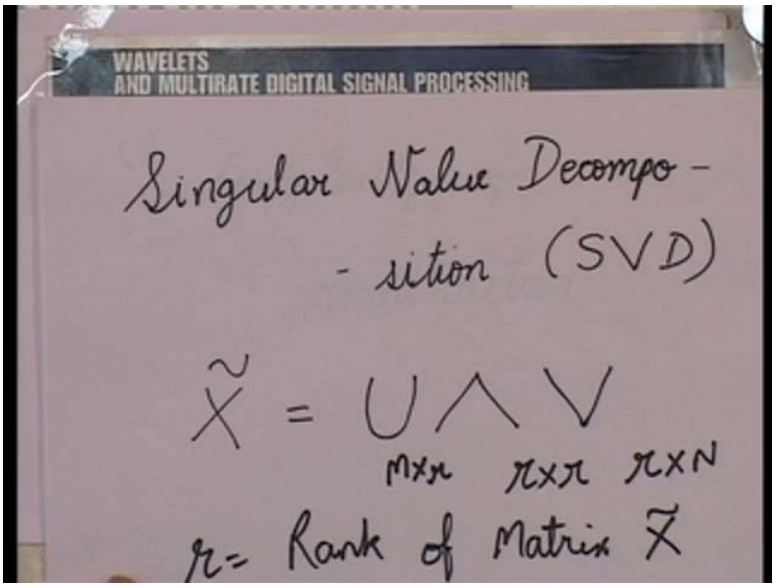
So, suppose X tilde is my whole data, so I represent that data in the form of a matrix, say of size M cross N. Suppose I store the stock prices of a particular company or say many companies, so let N be the total number of stock prices that I store for a particular company, say if I store for 1 year, then N will be 365. Let M be the number of companies of which I am storing the data, say each row of this matrix represents the stock prices of that particular company, this is how my whole data is represented; now what I need to do, I need to do three things.

(Refer Slide Time: 08:06)



First of all, I need to efficiently store this data, how can I efficiently store this huge amount of data? Say for example if I have data say 1 decade, say N will be equivalent to 1 decade and n will be say 100 companies or 200 companies, so efficient storage is very important. Secondly, how can I retrieve the data efficiently? And thirdly, if I want to add or modify certain things in the data how can I easily modify it? These are the three things, if these three things could be done efficiently, then we can using wavelets, then it will be a very good tool to the represent this data.

(Refer Slide Time: 09:02)

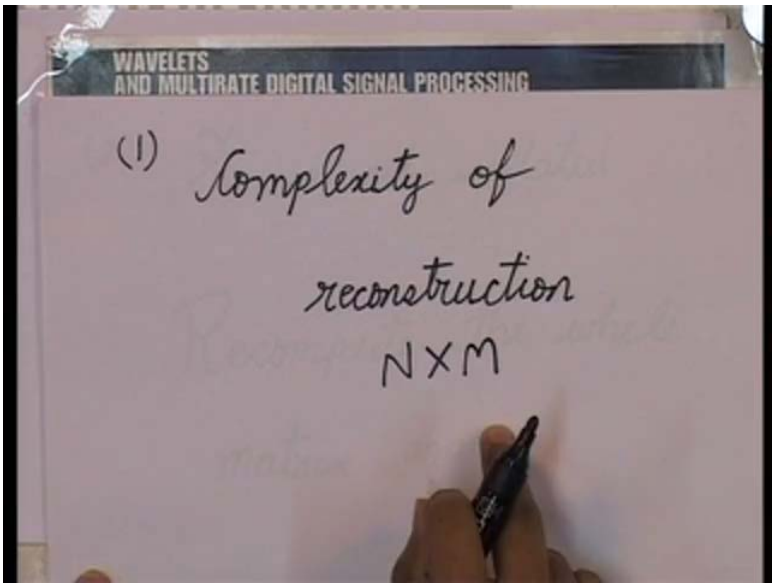


So, first of all, let us look at one of the methods which are already used in data mining, namely singular value decomposition method. What is done in this method is, I have a data say X tilde, I represent it in matrix form, U which is a common orthogonal matrix of size m cross r. This is the diagonal matrix of size r cross r and this is again a row orthogonal matrix of size r cross N, where r is the rank of matrix X tilde. So, I am representing this data in terms of three matrices, now instead of storing the whole data, I am storing these three matrices.

So, if r is small, I am already saving on that storage capacity of my data, which I need to store m cross n matrix; now I am storing just three matrices n cross r, r cross r and r cross N of which one of them is a diagonal matrix. Now, I have U diagonal matrix and V, suppose I want to extract the data of a particular company, say a particular row of this matrix X tilde, let that be represented by X. So, to extract the particular row, that means, to extract the data of a particular company, the complexity required is of the order of size of V, because multiplying these two, say this is a diagonal matrix, multiplying these two I will be able to extract a row of a particular matrix.

So, to extract a particular row at a complexity required is of these size of V, but this size of V is r cross N and N is huge in our application, say a data of a decade, say still the complexity is huge. So, the complexity of reconstructioning singular value decomposition is huge.

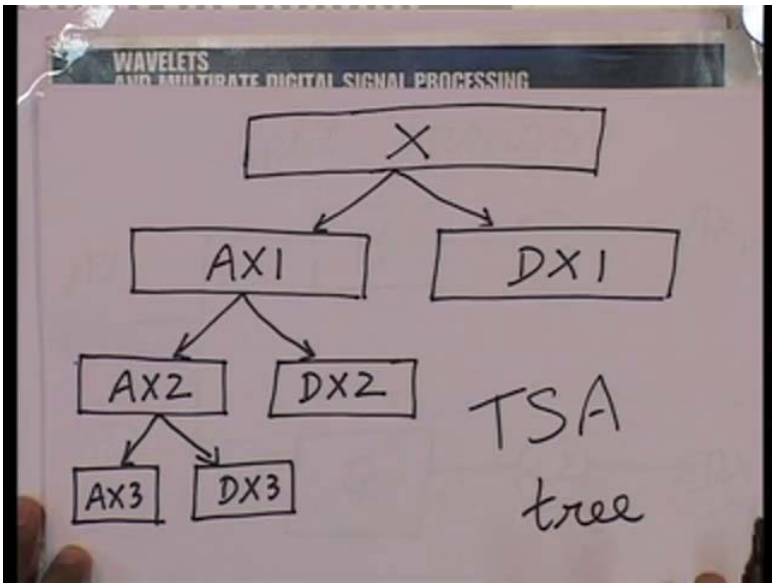
(Refer Slide Time: 11:07)



Now, one can argue here, let us reverse the order, instead of storing it as M cross N, why not store N cross M? Because M is not that huge, but the problem here is, now if you want to extract particular company's data, you need to extract the whole column rather than a row. The complexity of extraction of a row is of the order of size of V, not of the column. So, again if you want to extract column, you your size is equivalent to the complexity of V, so you are not improving your efficiency by reversing the order.

Further, one more disadvantage in this method is, if I want to modify the data, I need to recompute all the three matrices again, which is not the case with wavelets as we will see. I would just like to warn the audience here, that this does not imply that singular value decomposition method is not a good method. In fact, it is used extensively, but in this case, since our N is huge and since we require frequent updation, wavelet has a upper hand over this method.

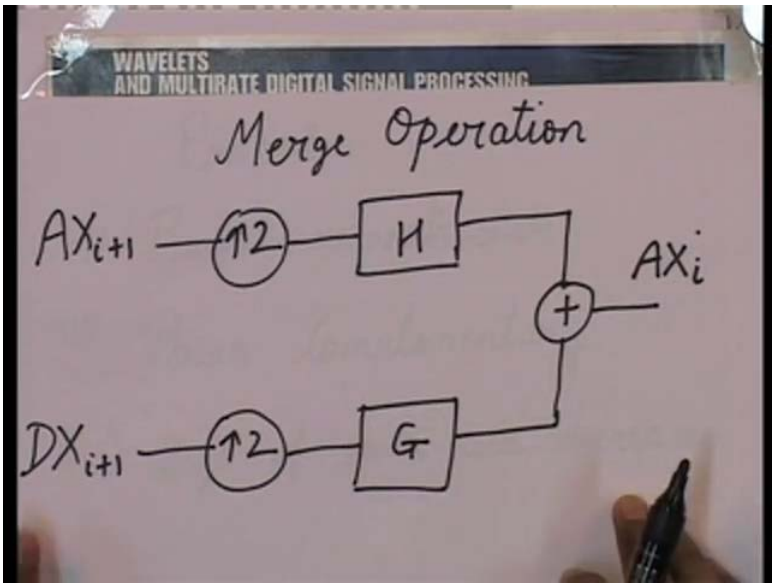
(Refer Slide Time: 12:18)



Now, what is wavelet? How are we going to store the data using wavelets? X is one row of that whole matrix X tilde, I am decomposing it into two sub spaces, approximate sub space and a detail sub space. The approximate sub space is also decomposed into two, one more a second level of decomposition, approximate sub space and a detail sub space, this is called as a TSA tree, this is Trend Surprise Abstraction tree; this stores all the trend data and this stores all the surprise data.

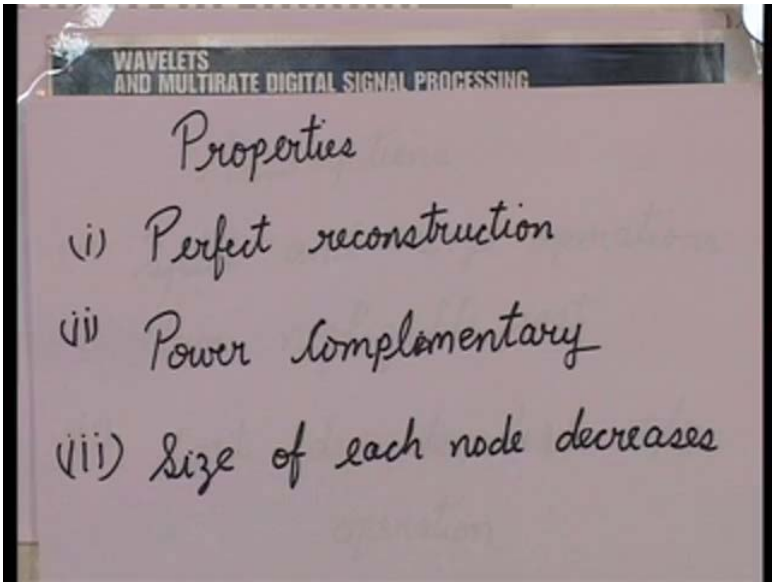
So, wavelets naturally store, decompose the data into forms trend and surprise, you do not need to abstract something, because wavelets naturally can decompose any data into trend and surprise form, this is the biggest catch here in this type of application. Now, how this this splitting and if I want to merge this, merging take take place.

(Refer Slide Time: 13:15)



This split operation is very easy, if I have a data A X i, I just pass through a low pass filter and high pass filter, down sample by 2 and I get the next level of sub space. So, this is a split operation, this is generally done in the course, I need not emphasize on this. Similarly, the merge operation is just the reverse of this up sample by 2, pass through the same filters and then add, so you go one level higher, so, this is the merge operation, the split and merge operations are easy.

(Refer Slide Time: 13:51)

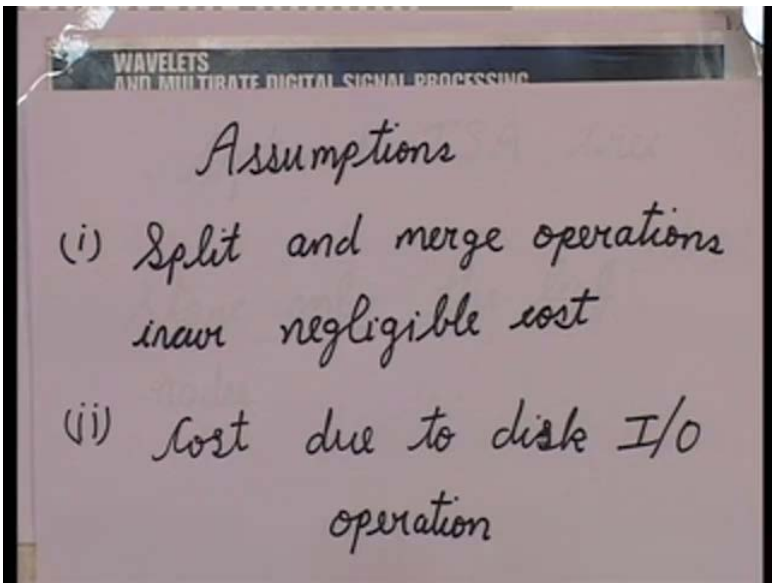


But the properties that this tree hold are very important, first of all, we can get perfect reconstruction. So, now, what I am doing here is (Refer Slide Time: 14:03) I am not storing this, rather than this I am storing this approximation sub space and detail sub space. So, I am storing this data rather than storing the signal and now I am saying, I can perfectly reconstruct this original signal from this data. And this is very important, perfect reconstruction is very important if it is not so, then if in case I have a point query, say what is a temperature reading this day? How will I get using that approximation and detail sub space?

So, perfect reconstruction is very important which is the case with wavelets, this we have already studied. Again, the power complementary properties also very important, because on decomposition, we are still preserving the power and this is also done, the third is very important, as we move down the tree (Refer Slide Time: 14:52), the size of each node, these are called as nodes is decreasing; so if the size of this signal is say N, this will be N by 2, N by 2 and so on.

This will be N by 4, N by 4, N by 8, N by 8, so size is continuously decreasing, so that is that is very important that we will see how. Secondly, these nodes are called as leaf nodes and we will see there, there instead of storing all the nodes, it is appropriate to store only the leaf nodes, now why this, since the size is decreasing by this point is very important.

(Refer Slide Time: 15:29)



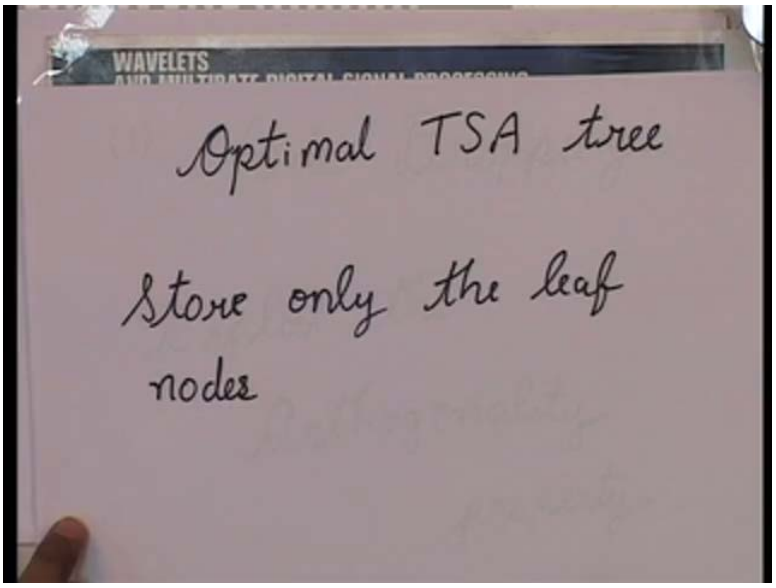
Because here, we are assuming that this split and merge operations do not incur any cost, they incur negligible cost, whatever cost is incurred is to extract the data or cost is due to the disk I O operation. So, the amount of data that you are extracting is actually incurring cost. So, cost is directly proportional to the size of the data and (()) our size is decreasing. If I want an eight level decomposition, what I need is just the approximation and detail sub space of the eight level (Refer Slide Time: 16:06) and the size of that is very small as compared to N.

So, the cost incurred is very less, just require a small and you can reach X, for example if I want a third level decomposition, I just need to extract a data equal to N by 8 and N by 8, I can perfectly re construct here, I can reach here and by that I can reach here X (Refer Slide Time: 16:30). So, just have a cost equal to 1 8 th the cost which I require by extracting the whole signal. Now as I pointed out, the leaf nodes are sufficient to give us the trend as well as the surprise details, how?

Suppose, I require a trend at the third level, so what I do is, I just extract this data and without using this, I do not perform perfect reconstruction, I reach this data by up sample and passing through a low pass filter (Refer Slide time: 16:46). Again without using this, I up sample and pass through a low pass filter and up I reach at this level and in this way I reach at this level. So, using a data equal to size N by 8, I am reaching at this level, without perfect reconstruction and I am getting the trend at the A X 3 level of decomposition. If I want the surprise data I go from here, go to A X 2, go to A X 1 and X; just here, I need to pass through a high pass filter instead of a low pass filter.

This is the synthesis branch (Refer Slide Time: 17:30), merging operation. Now, if I wanted this level, then first I need to extract these two both and I need to perform perfect reconstruction at this level and then I can go do approximation. So, this is (()) with small amount of posh processing of split and merge operation, I can find out the trend and surprise queries.

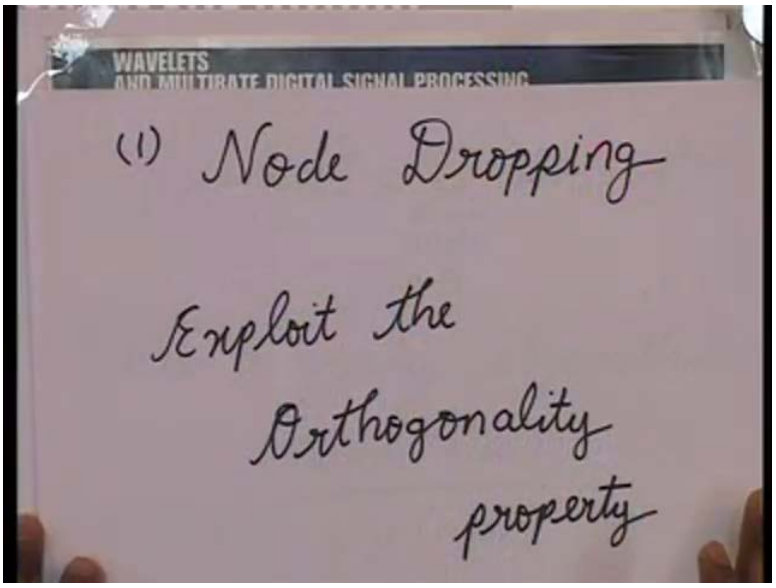
(Refer Slide Time: 17:57)



Now, the optimal TSA tree is that tree, which stores only the leaf nodes. Now, when I store only the leaf nodes, it is very easy to see that the total size of the leaf nodes is equal to the size of the original signal. So, I am not increasing my size by storing only the leaf nodes, I am not storing the whole tree, I am still getting all the information by using only the leaf nodes.

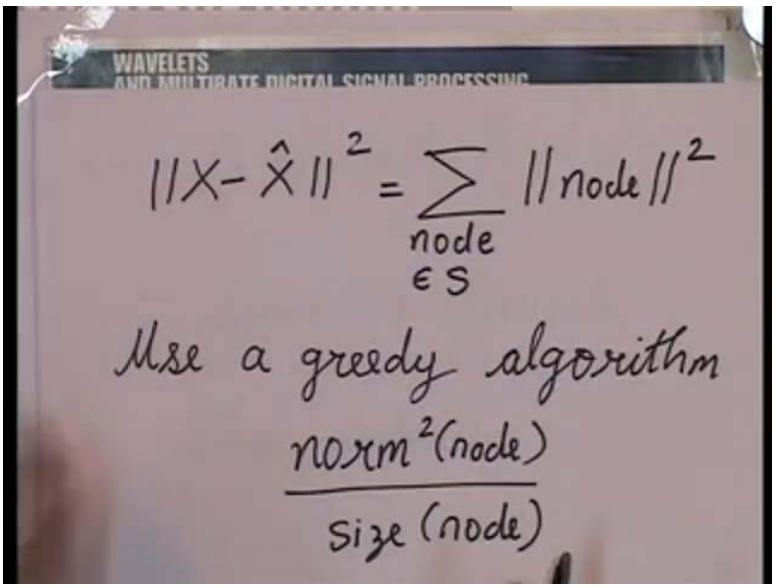
So, my optimal TSA trees that tree which in case a minimum cost and a minimum storage. And by storing the leaf nodes, I am incurring the minimum cost as well as a minimum storage. Further, if I introduce one more node which is not a leaf node, I will improve of performance, but that is also not difficult to prove that the performance increases just marginal. So, it is no points to store any other node accept for the leaf node. Once we have seen that even the retrieval is cost efficient, and even storage is cost efficient, can we improve more on the storage by reducing further on the leaf nodes? If I remove some of the leaf nodes, can I still get a almost accurate results?

(Refer Slide Time: 19:05)



Yes, wavelets as very good at compression, you compress leaf nodes and still you can efficiently store the data and improve your accuracy. I would not say improve your accuracy, but you are not compromising more on the accuracy. So, one of the methods is node dropping; at in node dropping, we are exploiting a very important property of wavelets which is the orthogonality property, what does that say?

(Refer Slide Time: 19:31)

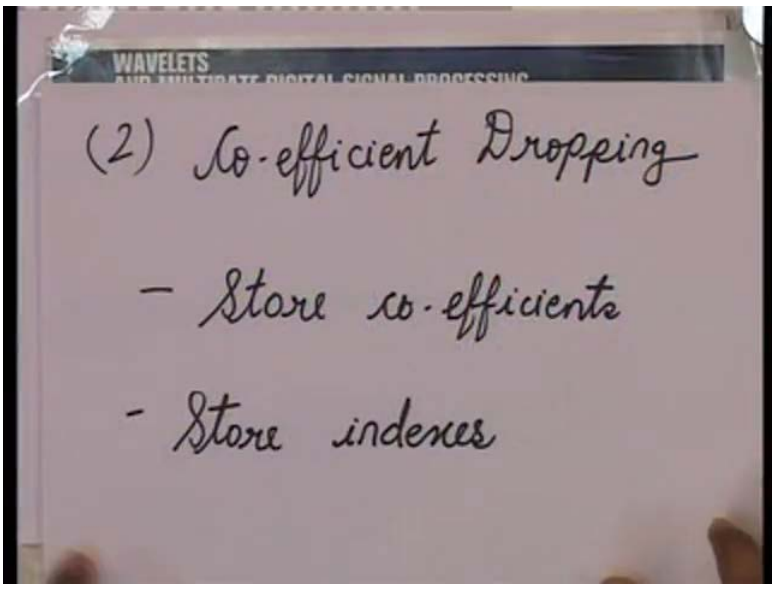


Suppose, I drop one of the leaf nodes, say d x 3 and I reconstruct the whole signal using the other leaf nodes and let that reconstructed signal be X cap. Now, if I find the norm, norm square in the error where X is the original signal and X cap is the reconstructed signal by removing one of the nodes, and then because of the orthogonality property, norm square is equal to the norm of the node which we have removed.

So, the error is actually the error in the coefficients which we have already removed. So, S is the number of nodes which we have removed, this is true only because of the orthogonality property of wavelets which is not the case always. So, if this is true, then we can use a greedy algorithm to determine, which are the nodes significant in the data? So, what I calculate for each node, each leaf node, I calculate the norm square of that node and divide by the size of the node.

And if I calculate the maximum of all and this is the most significant node which I find. So, I store that node, then again I store the second batch node, again I store third batch node and I reach a particular stage and if I find that I have already completed my disk space, then I remove the remaining nodes. So, this method is called as node dropping, but here there is a slight disadvantage in this method is that, if the node which I have removed contains some outlets or some important information, then that is lost which is not the case with coefficient dropping which we we shall see.

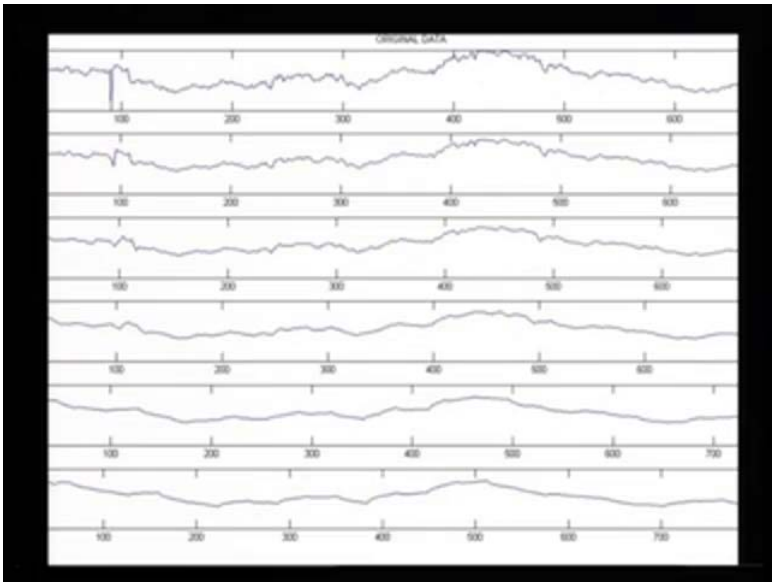
(Refer Slide Time: 21:12)



In coefficient dropping, I store all the leaf nodes as a sequence and each coefficient whichever are the significant coefficients; I store the whole significant coefficients rather than the nodes. And since, I am storing coefficients, I need to store the index of the coefficient also, as to this belongs to which node, since now for every coefficient I am using double the memory, storing the coefficients as well as storing the index, I have to just check how many coefficients I have used of a particular node.

If I am using a memory greater than that node size, then it is better to store the whole node rather than storing the coefficients. If I am not storing any of the coefficients of that node, I will remove the whole node. And if this size is less than the size of that node, then I will store this, these coefficients and this is called as a coefficient dropping method or hybrid coefficient dropping method. Now, let us the see the results what we have got here. So, I will just switch over to the slides to see the results, here these are the group members Kunal Shah, myself and Arka Choudhury.

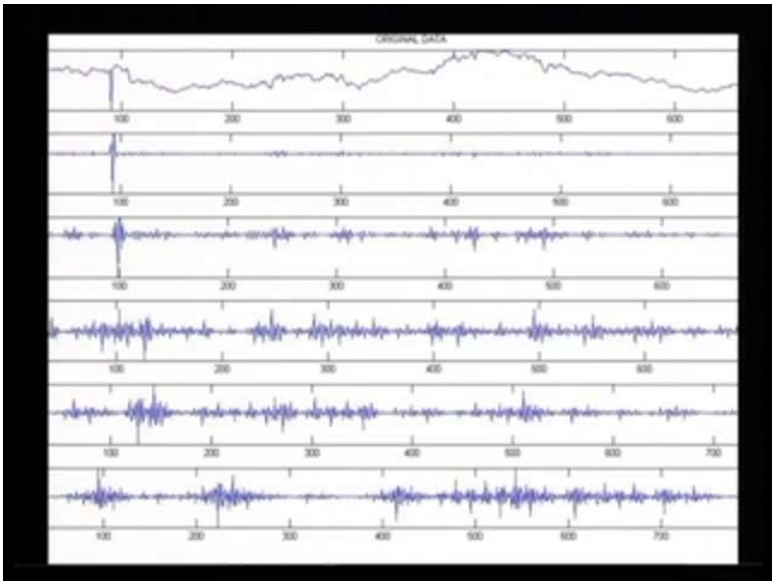
(Refer Slide Time: 22:23)



These these are the details of the trend analysis, this is the first graph is the original data, it is taken from yahoo stock market and it is data of SBI, State Bank of India of the two years. So, it has around 700 coefficients, these are the trends or the decomposition at various levels, the second graph is a two day decomposition, the third graph the four day ,fourth graph the eighth day and so on. And as you see, as you are going down the averaging this increased, so these are the trend analysis. If I see the last level and if you

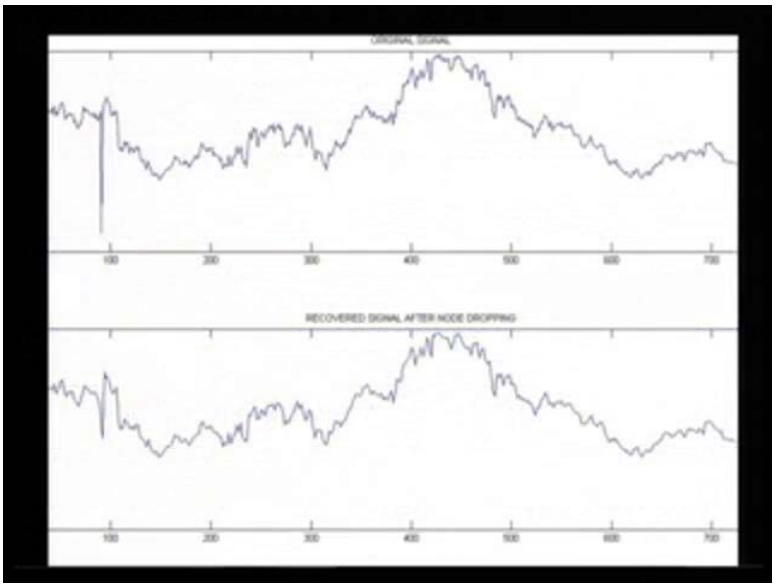
ask me what is the average our particular eight level data, then this is the eight level data average which you are seen, it would be more appropriate to show to interpret it on surprise analysis.

(Refer Slide Time: 23:18)



If you see on the surprise analysis, here the first graph shows the, there is a small out layer in the original data which is more prominent in the secondary average, in the secondary surprise as you can see. But as you go on increasing a decomposition level, you can interpret the data this way, the surprise which was looking prominent on two day is not looking as prominent as you increase the number of days, it is averaging out. In fact, you might see that by inspection from the original data you could very well observe that surprise, but as you go down the level that surprise is actually average down. So, if you look at the last level, there are many surprises which you can see in the previous data which by inspection you cannot encounter, only by decomposition you can encounter that here, here also there was a surprise component, here also there was a surprise component (Refer Slide Time: 24:08).

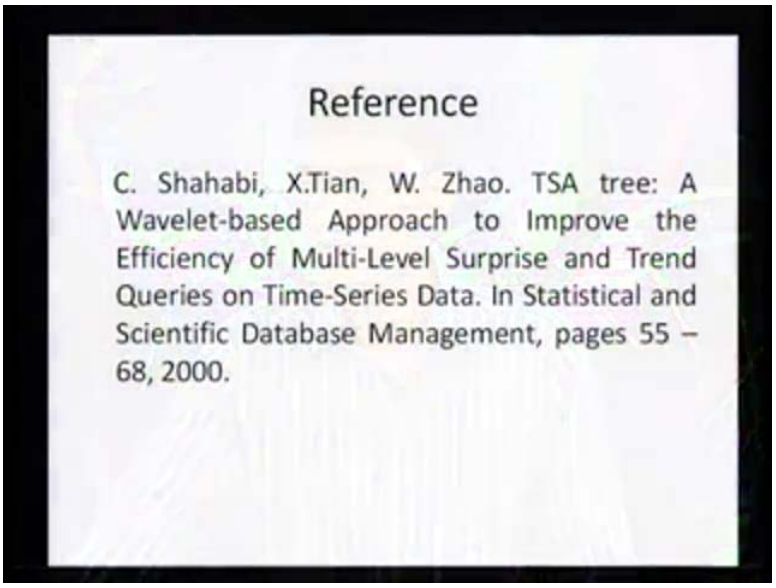
(Refer Slide Time: 24:13)



Now, this is a node dropping method, the above signal is the original signal and this is the recover signal after node dropping and we can see, there it is almost the same. But the disadvantage here is that we have missed that out layer which is seen that impulse out of which is seen in, we have missed in the node dropping and we have removed 300 coefficients in this, out of the 700 coefficients and this is the results. So, we we have removed almost 60 40 percent of the coefficients, retained only 60 percent. But in coefficient dropping method, as you can see that out layer is retained even though we have removed 300 coefficients.

In the trend analysis, we have used Do bash wavelet and we see that as the, as we move above the family, the only thing that is changing is the averaging increase more as you increase the filter length. So, as you increase the improve improve, if go on increasing from drop 2 to drop 4 to drop 8, the averaging is increased, this was all about application of wavelets in data mining.

(Refer Slide Time: 25:19)



The reference paper is the following, is as seen on this slide, I have just read this paper and interpreted on, as a student what I can interpret from this paper? So, my observations, my interpretations from this paper, I have just shown in this application of assignment.

(Refer Slide Time: 25:40)

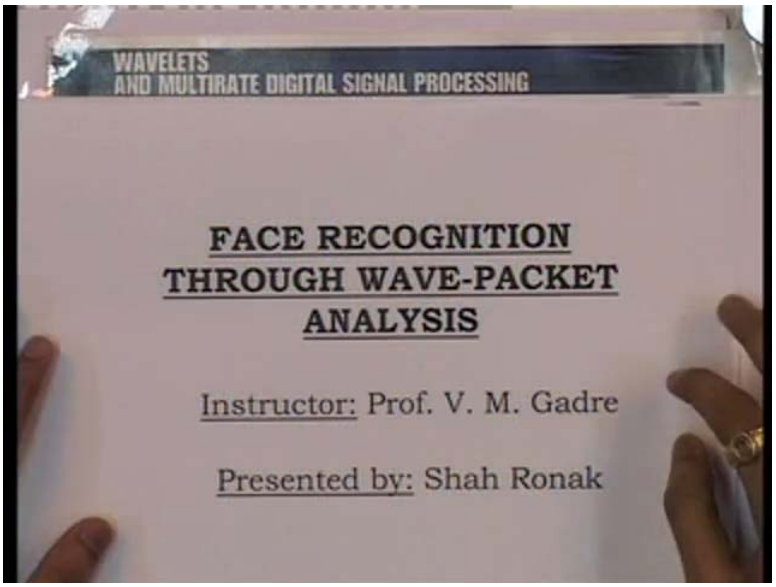
And this is the reference which I have taken for stock prices of SBI from yahoo stock. Thank you; I hope you enjoyed the applications assignment.

So, that was the very interesting presentation from Kunal Shah, on behalf of Kunal Shah and Arka Choudhury to go which worked on the applications of wavelets specifically, they would looked at the application of the do bash family in representing data basis efficiently, who will notice that the idea of course, representation and intimately information was given different meaning in this context. So, the idea of course representation gave what is called the approximation information, which we call a x and we so called increment information gave you the surprises, the novel information at every scale. Now, one must also take note of the difference between the node dropping approach and the coefficient dropping approach.

In the node dropping approach, one notice that with that the same level of compression, some important information was lost; the sudden spike was not seen so clearly. When coefficient dropping approach is by used, then one could see that; that was retained rather well. So, in addition to the transform that is used, how one uses the data obtained in the transform is equally important in wavelet based representation; that is what Kunal's presentation has amply demonstrated. Now, taking further the presentations of students on what they have done for their applications, we have the second presentation which was very briefly introduced in the previous lecture, namely that on face recognition. Without taking away from his presentation, I shall now invite Ronak Ronak Shah to make his presentation here based on the application of wavelets in face recognition, so Ronak for you now.

As professor Gadre mentioned, the second application that we are going to look into today is on face recognition through wave packet analysis. So, Kunal mentioned an application in data mining and it was in one dimension. So, the data that he has was in one dimension, when we move from data mine to face recognition, the data that we have is in two dimension.

(Refer Slide Time: 28:36)

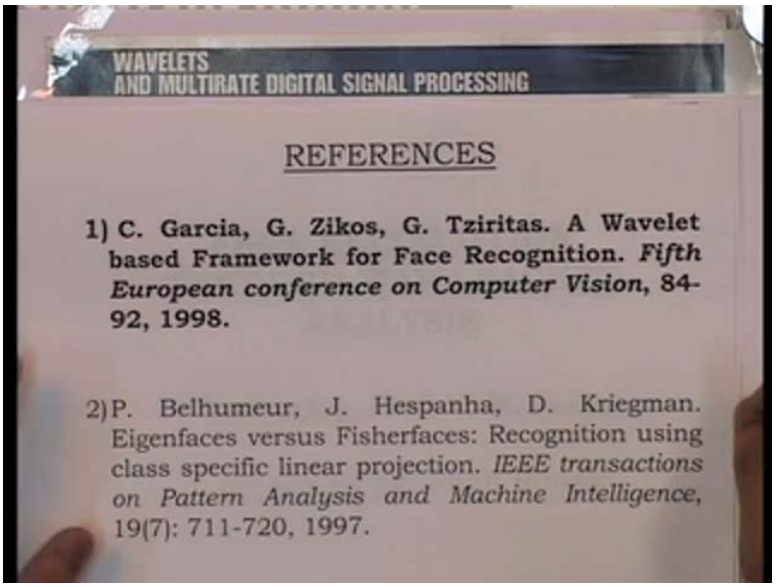


As I mentioned in the last lecture, there are two keywords here, one is face recognition and another is wave packet analysis. Before going into details, I would like to briefly summarize, what these keywords mean? I would start with wave packet analysis. Why we need wave packet analysis for the task of face recognition? As I mentioned in the last lecture, we need some kind of de correlation in spatial as well as frequency domain for the task of classification.

As we know, the task of classification can be better accomplished if we can do some kind of de correlation in spatial as well as in frequency domain. So, in that way, wavelet analyses, natural freeze into the scheme of things, but then why we go for wave packet analysis? Again, as I mentioned in last lecture, in wave packet analysis, we decompose not only approximation sub spaces, but we also decompose detailed sub spaces as we have seen in lectures.

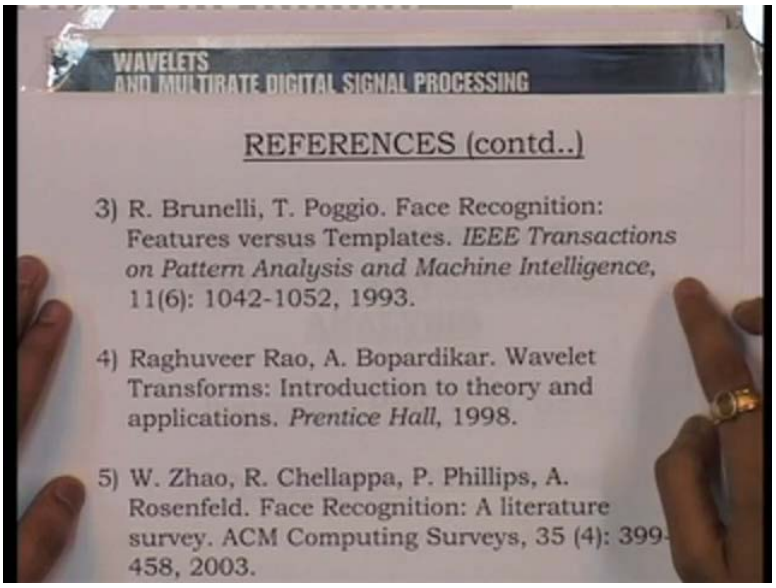
Now, when we do the task of classification, we need richer representation of the underlying signal that is we should not miss anything out from the signal, from the underlying signal and the underlying signal here is face that you have seen, so face imagines will be our signals. So, the task of face recognition can be can be accomplished by the wave lengths as well as wave packet analysis, but as I mentioned, we used wave packet analysis just for the richer representation and wave packet analysis also provides de correlation in spatial as well as in frequency domain, so that is better suited.

(Refer Slide Time: 30:14)



Now, before going into (()) and details, I would like to first provide references. So, the work that I am going to present here is based on the work done by Garcia, Zikos and Tziritas. It was research paper in European conference on computer vision and the work nicely by utilizing wavelet based framework of face recognition. Now, the work that I am going to present has been motivated by (()), but this is an extension as well as a normal description although work done by them. So, here we utilize a different database, namely the (()) database and the work provided by Garcia, Zikos and Tziritas was based on different database like farad data farad database as well as feces database. So, here we do not only provide extension to different database, but here the understanding is purely from a point of view of a student, how was student can look into this application and how this particular framework based on wavelet, it fits into the whole framework of face recognition, like what are the approaches that have been there the literature and how that fit in, how this approach fits into a particular framework.

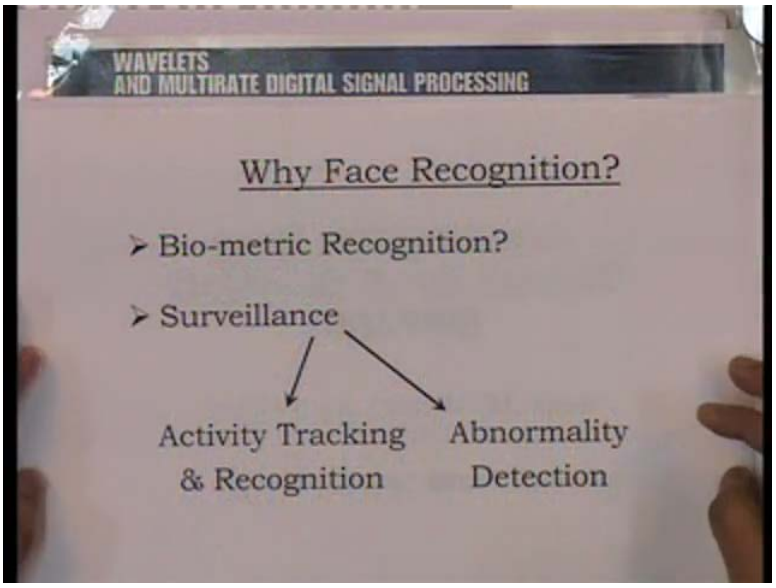
(Refer Slide Time: 31:28)



There are some other references, like the second reference here is based on Eigen faces and this is quite a popular approach. Also, there are some other references like the approach based on PCA, Principle Component Analysis, this framework has also been popular. The reference that I mentioned, the first reference compares this PCA based approach and Eigen faces based approach to wavelet based approach. The fifth approach here that is mentioned like by Zhao, Chellappa, Phillips and Rosenfeld, it provides a nice framework or nice literature survey on the overall work that is done in the task of face recognition. So, this is for references, now we shall moving to like, why face recognition is required.

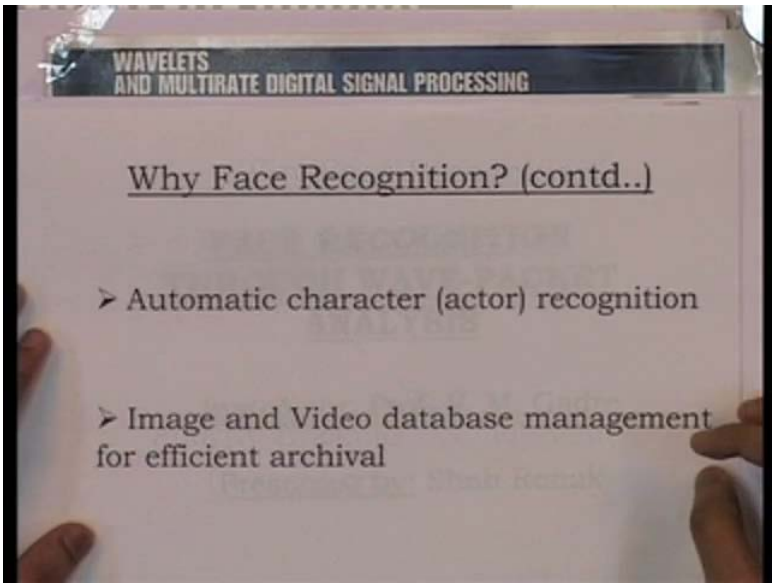
As I mentioned in the last class, face recognition can be required for biometric authentication. But as I mentioned, the face like the signal that is obtained from face, namely the image, a 2 D image, it can easily be morphed like someone can grow more stash or someone can go beard in a given period of time and that can easily be morphed, or someone can come of with sunglasses even. There are some other signals which are available, like retina or let us say, for that case fingerprints, they cannot easily be morphed and they can easily be utilized by biometric authentication. So, the task of face recognition for biometric authentication framework is limited, however we need face recognition for the task of surveillance.

(Refer Slide Time: 32:47)



In surveillance, one needs some area or region to be (()) and some generally in these kind of region, only a limited number of persons are allowed. So, the excess is allowed, also one might be interested in what that subject is doing inside the particular region. Also we have some (()) prototypes, like which kind of activities are allowed, someone is telling inside that particular region or someone is doing some kind of abnormal activity in certain region, you want to detect that is well. And generally, the queue for this kind of framework starts with face recognition. So, under the task of surveillance, we may do activity tracking as well as recognition and if we can do this faithfully, then we can also provide abnormalities.

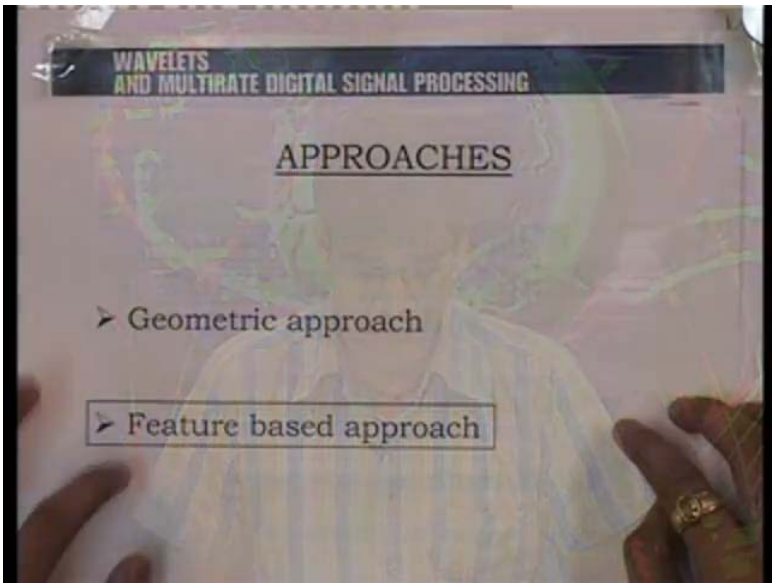
(Refer Slide Time: 33:31)



Not only this task, but also in tasks such as automatic character recognition in let say, movie clips. If there are some broken movie clips or let say some parts of the clips are available, or lets of a example you tube, then we can also perform automatic character recognition which actors are present in which scenes and we can also do efficient archival based on actors.

So, in you tube, when a large database videos and images are available, we can perform this archival task faithfully. So, we can classify of the showpiece like shows and serials based on what are the characters that are present inside those scenes. So, this is for the task of face recognition, why we required face recognition and we see the task of face recognition is crucial, so accuracy should also be crucial for that case.

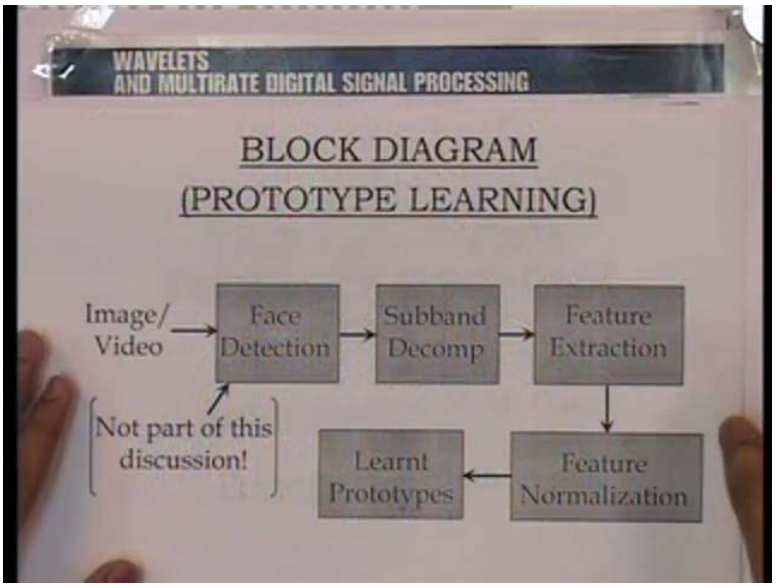
(Refer Slide Time: 34:20)



Now, the approach that I am going to mention falls under the falls under the approach of feature based approaches. So, there are two approaches that basically used for the task of face recognition, one is geometric approach, another one is feature based approach. In geometric approach, one goes on detecting the basic features of legs and nose or eyes, cheeks, chins, etcetera and generate features based on this this extracted features of nose, eyes and cheeks and chins.

But the problem here is this features are difficult to extract, because what we have in effect is just a 2 D image. So, why not to look at face image as just a 2 D image and we can represent in some other features rather extracting eyes, noses and other features, etcetera. So, the other feature that I am, the other approach that I am going to talk about is a feature based approach and approach that I am going mention falls under the category of feature based approach.

(Refer Slide Time: 35:19)



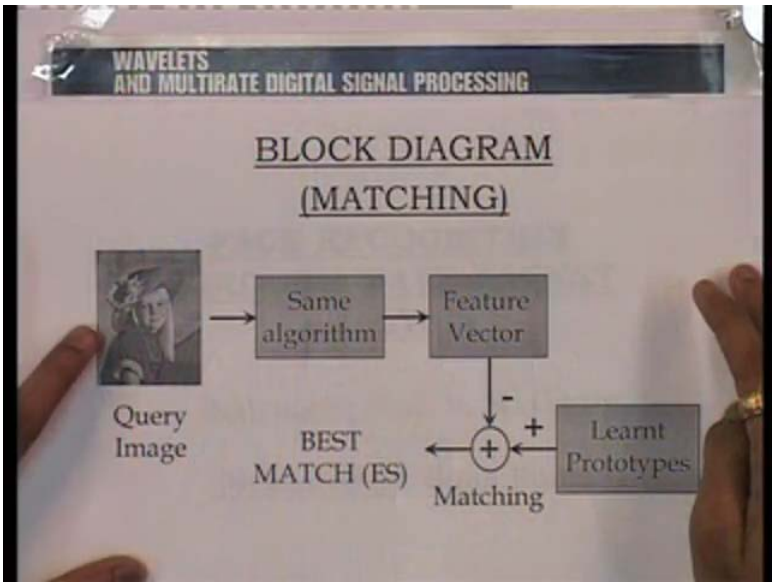
The block diagram, if we look at of the approach that I am going to mention looks similar to this, this is for prototype learning. So, given a image or video, one may detect faces, so this is a face detection, I will go at them. But this is not part of the discussion, because as I mentioned in the last lecture, these two things can be decoupled easily, because the face detection, one looks for the features which are very similar across all the faces so that you can faithfully detect, what are faces that are available in the image and let say for example, a video, where as in case of face recognition will look at features which are distinctive across faces for faces.

So, this can be decoupled and hence, we are not going to talk about face detection here, what we are going to talk about is face recognition. So, after extracting a region of interest in which faces are available, we can perform sub band decomposition and here we are going to perform sub band decomposition by using wave packet analysis. After performing sub band decomposition, we go for feature extraction, now this feature extraction is the crucial job, because we cannot involve each and everything into be inside a feature vector, that I am going to discuss next.

We can also perform feature normalization, because we do not want one feature vector for each image, rather we want feature vector per class, so if you have, let us say 6 to 8 images per one subject, then we want to compact feature vector for each subject and not for each image. So, feature normalization can also be performed, this will be basically

give as learnt prototypes and this learnt prototype can be stored in memory and can be accessed in future when a query image comes to us, this is the black diagram of matching. So, if we have learnt prototypes with us in which feature vector of different classes are stored, then we can perform matching.

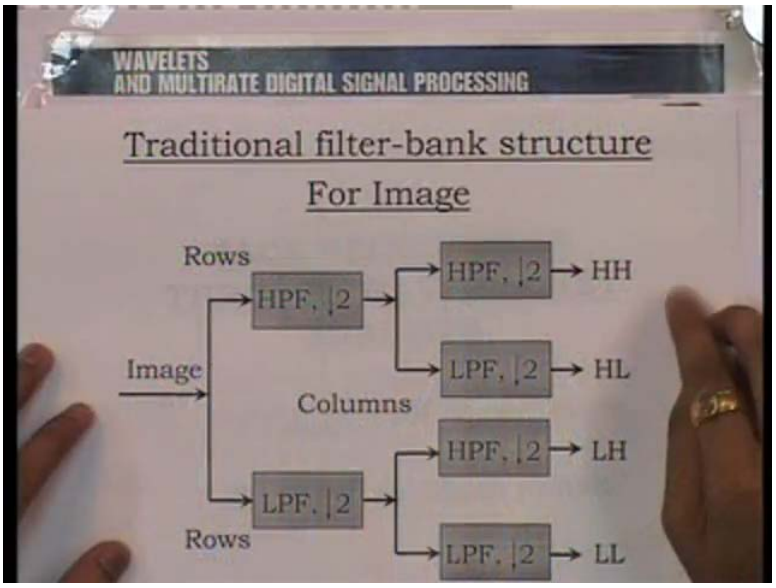
(Refer Slide Time: 37:04)



So, given an image like this, a query image, we can first do face detection and then we can basically apply the same algorithm, we will end up getting the feature vector.

And we gave learnt prototypes that we can access from memory and we can utilize some distance metric or some similarity to come up with different matches. So, based on application, we can give output as one image or multiple images. Now, there can be two kind of applications for face recognition, one minute based on content based image retrieval, one image is given as input. In content based image retrieval framework, we need all the images that are there in the database to be expected which match to the query image and in another application, we can perform a simple, traditional classification in which an image is given and we just want to know, to which class it belongs to; in that case only one match is required.

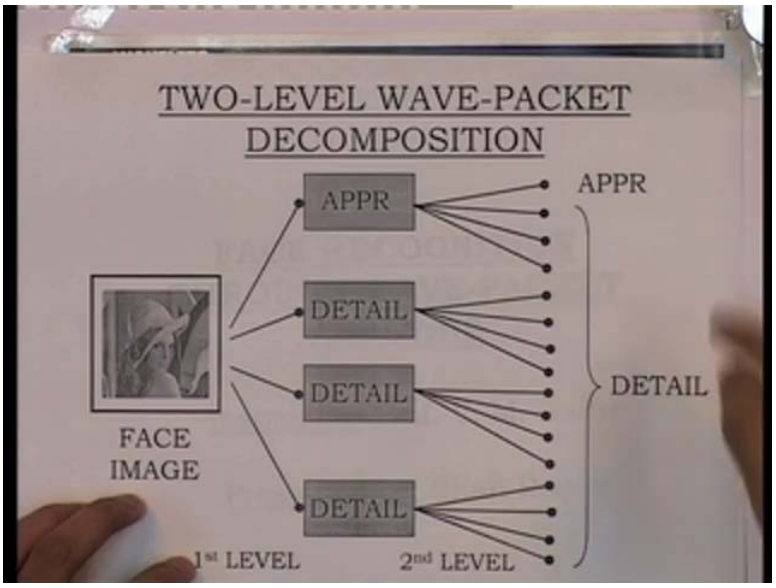
(Refer Slide Time: 38:04)



Now, how do we perform decomposition? Let in case of wavelet analysis like this application like, so this slide provides how we can perform decomposition of a 2 D image, this differs slightly from a 1D signal, because now the image is 2 D here, so this is a 2 D signal. So, we need to perform of filtering in rows as well as in columns and down sampling will also be performed in rows as well as in columns, so first we perform filtering across a rows and we put down sampling across rows.

After getting images here by performing filtering as well as down sampling across rows, we can move to columns, we can perform high pass filtering and low pass filtering and down sampling across columns and we can get four sub images. So, this is LL is the approximation substitutes here, because it is passed through by two low pass filters and other filters like other sub images that required to be pass from high pass filters, so they contain high frequencies in effect, but this effects from wave packet analysis.

(Refer Slide Time: 39:05)



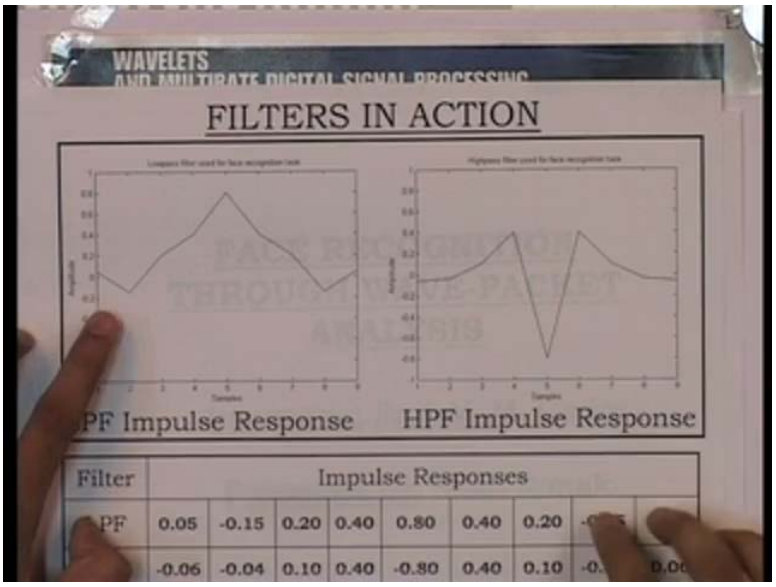
In wavelet analysis, we move from first level to second level of decomposition by only decomposition approximation subspaces, whereas in case of wave packet decomposition, we decompose detail subspaces as well. So, this is the first level of decomposition, this is the approximation subspaces and these are detail subspaces, then we denote only decompose approximation subspace, but we also decompose detail subspaces. So, we will end up getting 16 subspaces here in which one is the approximation subspace and others are, other 15 are detail subspaces.

So, this approximation subspace will contain low frequencies as derived and this detail subspace will will only contain high frequencies in this order. So, this last detail subspace will contain the most high frequencies. So, we get some kind of decomposition in spatial as well as frequency domain and this also provides de correlation. So, now, we can generate feature vectors, but before going into that, we need to move into which kind of filters we are utilize for this purpose. Now, the important advantage of using this wave packet application is, we can utilize any filters here.

When we go for face recognition, we do not want synthesis of perfect reconstruction to be done, we only want our analysis filters to be good so that they can provide de correlations spatial as well as in frequency domain. So, here without worrying about orthogonality property or any other property, we can generate those filters with provide

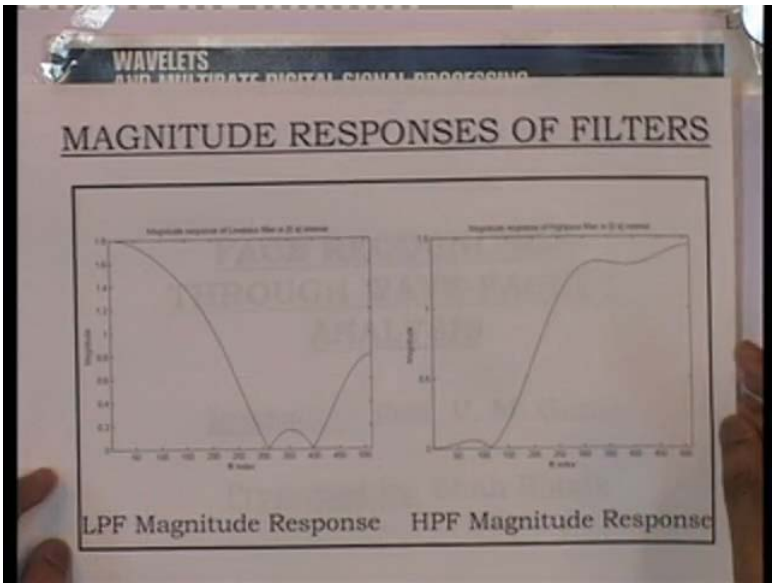
very good feature vectors for our classification task. So, in practice, one can go for empirically searching filters which are best suited for this application of face recognition.

(Refer Slide Time: 40:45)



And here are the features which were to the empirically suited to the task of face recognition, and these are the impulse responses of low pass filter and high pass filters that are used for the task of face recognition, this filters are discrete in nature, but for ease of visualisation, they are connected like the two intermediate samples are connected by straight lines. So, here we can see the impulse responses of the low pass and high pass filters, one might also worry about how the local frequency domain and this line basically shows the magnitude responses.

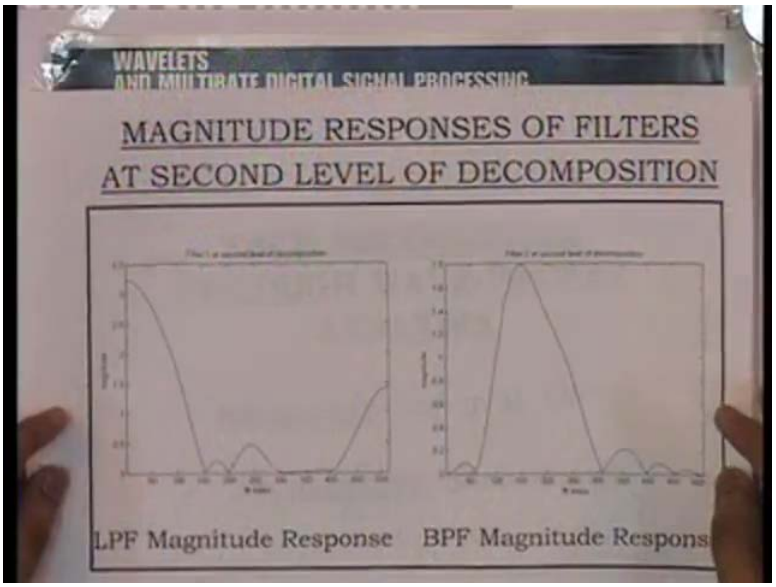
(Refer Slide Time: 41:12)



So, this corresponds to crudely a low pass filter and this corresponds to a high pass filter. So, the filter that way views they do not, they are not so much different from what we have seen in the course, they are basically low pass filters and high pass filters, but this magnitude responses were found to be better suited to face recognition task. So, when we talk about first level of decomposition, this high pass filter and low pass filter were found to be better, but we are decomposing this to this second level, because after second level, generally the images that have for face recognition they are quite small in nature.

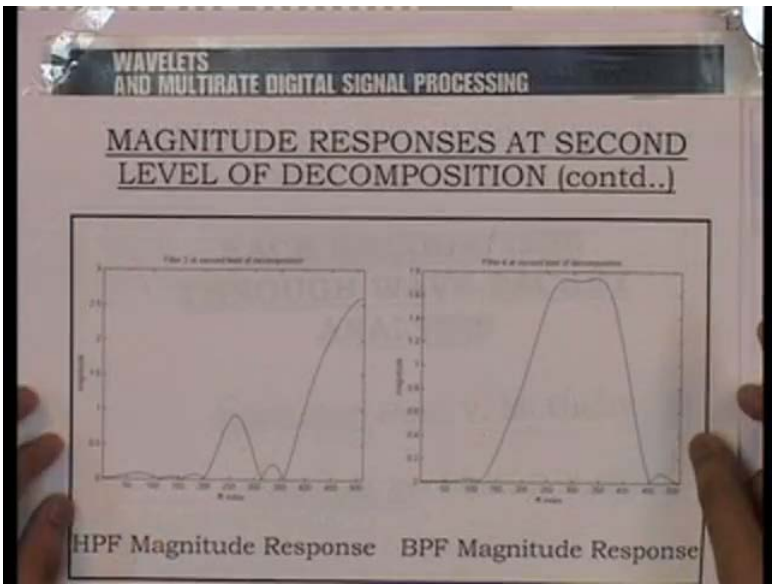
So, as we go on decomposing this images, they become smaller and smaller. So, becomes really tedious after second level of decomposition to handle this small images and also the localization that we obtained in time domain, it becomes really crude. So, they are not of much importance, they do not need much information. So, we go up to second level of decomposition. When we look at this wave packet analysis in second level of decomposition, we obtained four filters and I am going to present the magnitude response is these of four filters.

(Refer Slide Time: 42:21)



So, when we decompose an approximation subspace, we obtain magnitude responses of this low pass filter and this band pass filter, here we can see that this is crudely a low pass filter again, but this cut off is quite range in down and here this is a band pass filter, but this mostly emphasis the low frequencies which are available, because we are decomposing the approximation subspace.

(Refer Slide Time: 42:48)



Also when we decompose a detail subspace as is required in wave packet analysis, we end up getting these two filters, again one is a high pass filter and one is a band pass

filter, but this band pass filter emphasis wave frequencies rather low frequencies as we saw in the previous slide. Now, this is this filters are ok when we talk about one dimensional signal, but here we are talking about two dimensional signal, but the importance characteristic of this filters are, they are separable in nature.

So, as you apply in one dimension, you can apply second dimension and we end up getting 16 filters rather than 4 filters as in one dimension case and but the ease of visualization in one dimension helps us to talk about one dimension. And I will not talk two two dimension impulse responses and magnitude responses, because they are just natural extensions as well as they are difficult to visualize in nature rather than 1 D signal.

(Refer Slide Time: 43:39)

So, given a (()) image let say for example, we can end up getting this kind of subspaces where this is the approximation subspace, we can easily visualize this because this content low frequencies and the information here is more compared to other detail subspaces. These are seven other detail subspaces, important thing to note here is this detail subspaces and approximation subspaces are quite small in nature, but just for ease of visualization have been zoomed into, also like this detail subspaces are bipolar in nature like they contain negative as well as positive values.

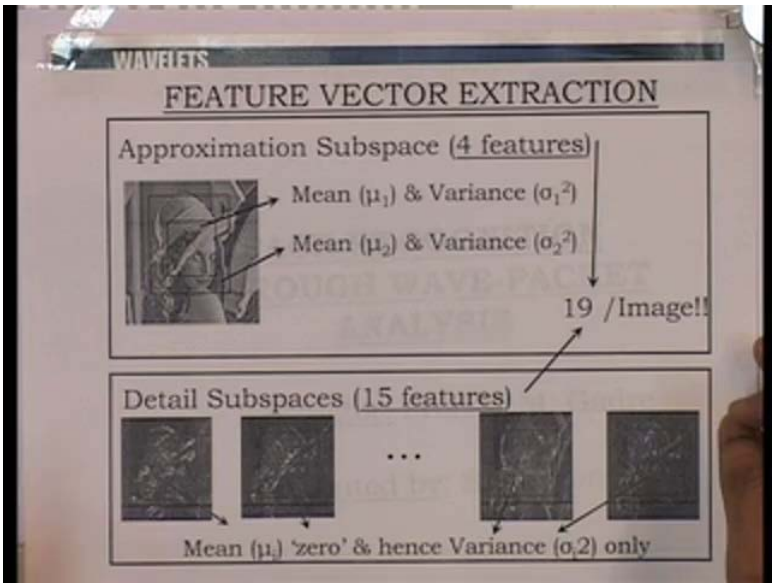
So, just for visualization, they have been taken as absolute, the values are absolute here, the other subspaces are also detail subspaces and we can visualize here (Refer Slide

Time: 44:20). So, those were it and these are again 8 other detail subspaces, these are all 16 detail subspace that we are going to obtain, here we can see that some of the features like eyes are very clear in the high frequency subspaces and other subspaces contain the overall shape of the face image. Now, after getting this subspaces, 16 subspaces namely, we can go on to do feature vectors extraction. Now, the trouble with feature extraction, see here the important thing is to extract those features which are relevant to face images, one might say that all the features, all the pixel values that are there in 16 sub images are important to us and go on to (()) all the values that are available to us.

But the problem doing this is, this feature vector will be quite long and if you want to do a classification task, this becomes increasingly difficult, because if you want long feature vector, we better have long number of images, large number of images, then and then we can go on to do a better classification. So, rather than going into this, we can provide a compact representation of the faces my providing, one we have doing this is to go for moments. So, first order moment or second order moment we can go for mean and variance basically.

So, we have 16 sub images, we can go for mean and variance instead of in each sub images. So, we can extract a thirty two dimensional feature vector, but here if we look at detail subspaces, then they have zero mean, so we do not need to capture that. So, we can basically go on to represent seventeen dimensional feature vector for each face image, but if we can see at approximation and detail subspaces, they are different in nature. In approximation subspace, we have more information (()) get us for handling approximation subspace differently for rather than detail subspaces.

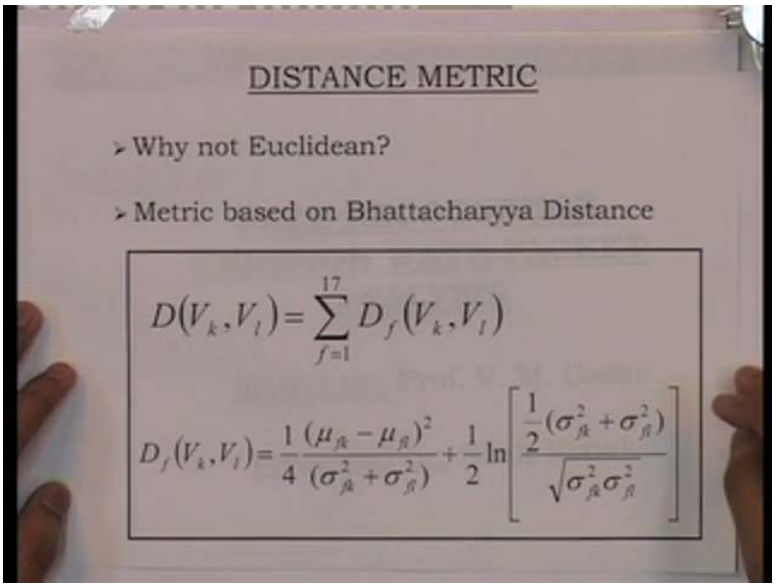
(Refer Slide Time: 46:05)



So, if we look at this approximation subspace, what we can do is we can bounding by two boxes, in one box we can basically go for a high dimensional features like, what is the exact shape of the face or what are other fringe information that we have in that bounding box and we can go on to get us for mean and variance for that bounding box, we can taken other bounding box in which other information related to shape, like face of eyes and nose are related. So, we can extract mean and variance in that bounding box.

So, we can extract 4 features, from approximation subspace and we can extract variance is in each detail sub image and we can extract fifteen dimensional feature vector from here. So, four dimensional feature vector from approximation subspace and fifteen dimensional feature vector from a detail vector subspaces, we can extract nineteen dimensional feature vector for each image. Now, given a task of classification, we can go on to extract nineteen dimensional feature vector for each face image and in effect, we can do some feature normalization so that we can end up getting one feature vector for each class and after then we can compare the feature vectors that we have in data base with the feature vector of the query image.

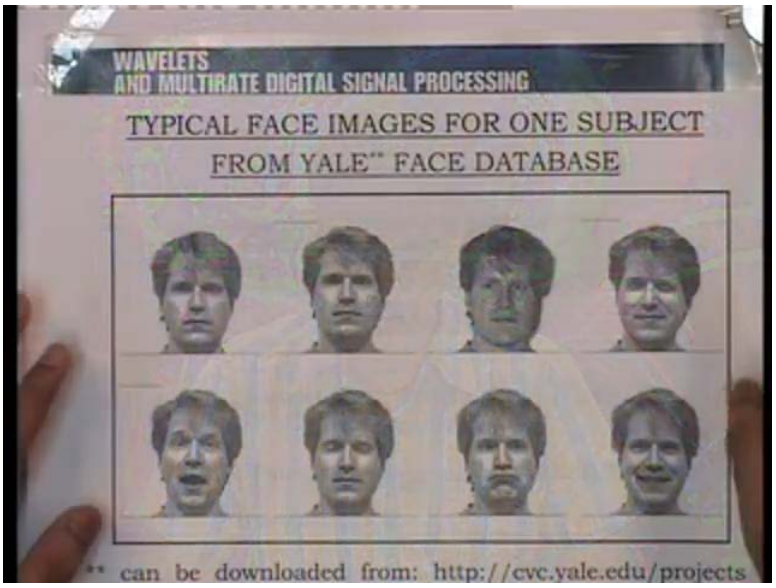
(Refer Slide Time: 47:23)



And we can do some kind of, we can utilize some kind of distance metric, but the important thing here to note that is we have utilize mean as well as variance. So, we have information related to probability density functions of the underlying feature vectors, now if we utilize Euclidean distance as the feature vector as distance matrix, then the trouble is Euclidean distance assumes all the dimensions to be independent of each other. So, it is it does not cut off for the probability density function that we have, rather than we use Bhattacharyya distance that takes care of probability density function that we have, it basically goes on for mean as well as variance. So, we have two different entities here, one for mean as well as one for variance.

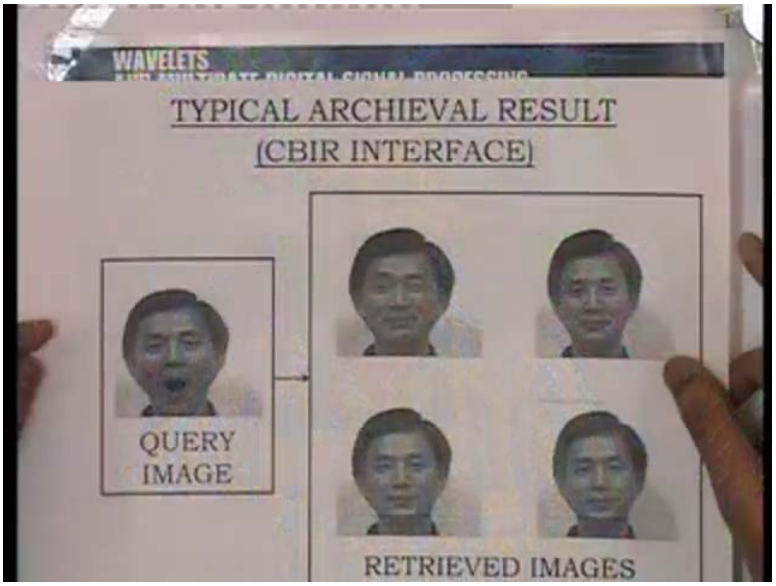
So, if we have the same probability density function, mean will be similar as well as variances will be similar. So, both these terms will varnish and we will get a zero distance the exact match, also Bhattacharyya distance get us for all the requirements needed for it to be a proper distance metric. So, Bhattacharyya distance can be utilized, but there is in nothing magical about Bhattacharyya distance, any other distance metric which get us for PDF's, underlying PDF's and of a feature vectors can be utilized for the purpose.

(Refer Slide Time: 48:36)



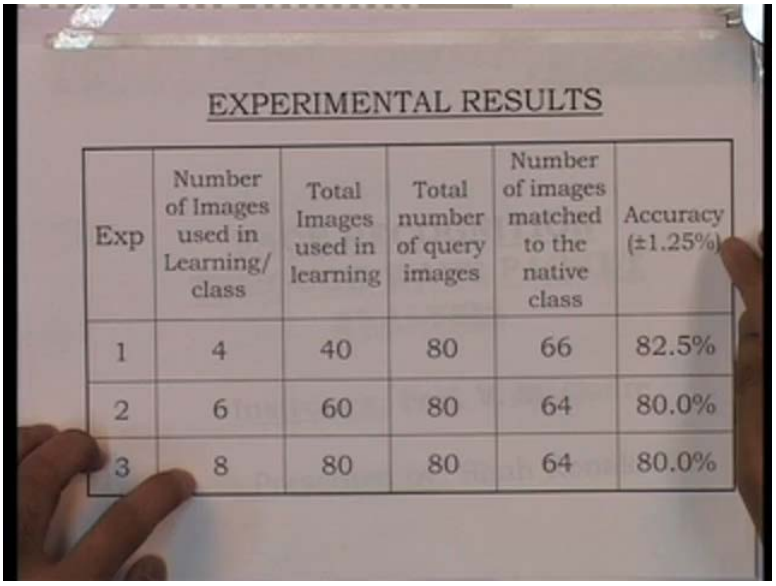
Now, we will quickly move towards results that we that, I have obtained by utilizing the Yale data base. Yale database is basically available for free download for non commercial use from this link and these are the typical faces that are available in Yale data base for one subject. Yale database basically contains 165 images for 15 classes and 11 images for class, this is these are the typical images that are available in Yale database.

(Refer Slide Time: 49:02)



I just provide a typical archival result that I have obtained from the CBIR interface for face recognition task. So, given an input like this, a query image with the surprise expression over the face of the subject, we could retrieve images similar to the subject like the subject is similar in all the retrieved images, but the expressions are different like one is blinking for example, the other is with happy look on the face, etcetera. So, this is the CBIR interface, but in order to quantitatively evaluate the algorithm, we need to provide quantitative results and this can be provided by traditional classification task.

(Refer Slide Time: 49:40)



So, these are the experimental results, three experiments were performed based on different number of images utilized per class for learning. In first experiment, four images per class were utilized by learning; in second experiment, 6 images were utilized and in the third experiment 8 images for utilized per class, so in total in first experiment we had 40 images for learning and second experiment we had 60 images for learning and third experiment we had 80 images for learning, all the 80 images were utilized as query images.

And the fifth column shows the accuracy that we have obtained; in first experiment 66 images match to the native class. In second experiment, 64 images and in third experiment, 64 images again match to the native class. So, the accuracy is around 82 82.5 percent to 80 percent within the accuracy of 1.25 percent. Now, before going before (()) too much of results I would like to remark on some of the things like this

experimental results can be enhanced like this 80 to 82 percent accuracy can be enhanced to 90 to 95 percent of accuracy if one goes from (()) learning techniques like for example, support vector muscles or one can probably utilized more number of moments like (( )) or third order moments, etcetera and this accuracy can be enhanced. With this with this remark, I would like to conclude this application, thank you very much.

That was a very beautiful presentation on the application of wavelets, specifically wave packets in face recognition. I would like to emphasise of few of the points that were made in the presentation, one was the distinction between face detection and face recognition. In fact, in the sense wavelets for the (()) wave packets are suited both to the detection and the recognition problem. In detection, one looks for commonality, work constitute general face; in recognition, one looks for specificity, what is the inclemently information in that face. So, the separation into common and the incremental information is again critical in context of both face detection and face recognition.

We looked at the beautiful the decomposition switch Ronak showed using wave packet analysis, we noticed the different kinds of features came out in different sub bands. Now, in fact, one can study that greater depths, what is being presented here is an indicative study, what different sub band show, you know Ronak also presented the frequency responses of the filter that the second iteration in wave packet analysis. And you notice again a confirmation of that theoretical discussion that we had when we carried out wave packet analysis in one of the previous lecture, we saw that when we decomposed the high frequency band, that is, when we followed high pass filter and down sampler by the analysis filter band again, there was a band in version.

So, high followed low, actually gives you the higher frequencies and high followed by the high give you the lower frequency, this should be noted as we notice that the expected, there was two band pass filters aspiring to become band pass filters between pi by 4 and pi by 2 and the normalized to omega axis, and pi by 2 and 3 pi by 4 on the normalized omega axis. And of course, there was a low pass filter with cut off pi by 4, there was a high pass filter with cut off 3 pi by 4, these are all these were all aspiring to become these kinds of filters, now we have seen two very interesting applications of wavelets in filters bands today.

There are several others as I have just mentioned a couple before we conclude the lecture today, one of the other important areas of the application of wavelets and time frequency methods is by bio medical signals, bio medical signals could be evolve evoke potentials, they could be magnetic (()) signals (()) signals, they could be signals obtained from cat scanning computerized action tomography scanning and many other imaging (()) useful in medical image processing today. There is one area in which wavelets and time frequency methods filter bands had been heavily employed.

We hope to have another presentation of an application of this kind in a subsequent lecture. Wavelets have been used also are proposed by used in digital communication, people have talked about building modulation and demodulation systems on wavelets, because they are nice time frequency atoms. Another area for the mathematician in which wavelets have found great applications is the use of wavelets in solving differential equations. At this point, I shall not say more, say to mention that such some of the (()) areas in which wavelets are used. We shall hopefully be able to see some of these applications in greater depth in subsequent lectures. With that then, we come to end of this lecture and we shall proceed to discuss something different in the next one, thank you.