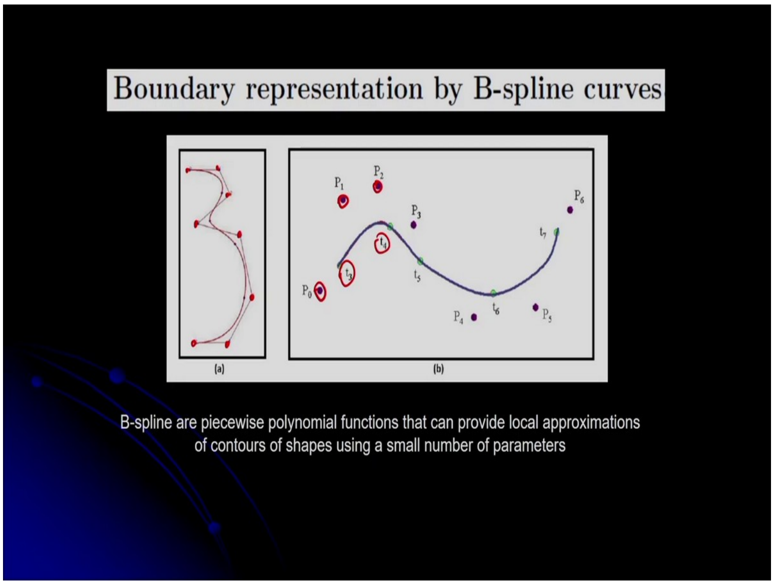
**Computer Vision and Image Processing - Fundamentals and Applications**
**Professor Dr. M. K. Bhuyan**
**Department of Electronics and Electrical Engineering**
**Indian Institute of Technology, Guwahati**
**Lecture-27**
**Object Boundary and Shape Representations - II**

Welcome to NPTEL MOOCs course in Computer Vision And Image Processing - Fundamentals and Applications. In my last class I discussed about the shape and the boundary representations, I discussed about chain code and the Fourier descriptors and also I highlighted the concept of B spline for boundary representation. The B spline is nothing but the piecewise function that can be used for approximating a particular boundary.

In B spline representation, I need some control points, some knot points and also I need the B spline function. So, by using this I can approximate a particular boundary. So, instead of storing the entire boundary pixels, I can only consider the B spline function and the control points for the representation of that boundary.

So, I can consider only the B spline function and the control points for representation of the boundary. So, now, I will be discussing about the B spline representation, so how to represent a particular boundary by using the B spline function. So, let us see what is the B spline function.

(Refer Slide Time: 2:00)



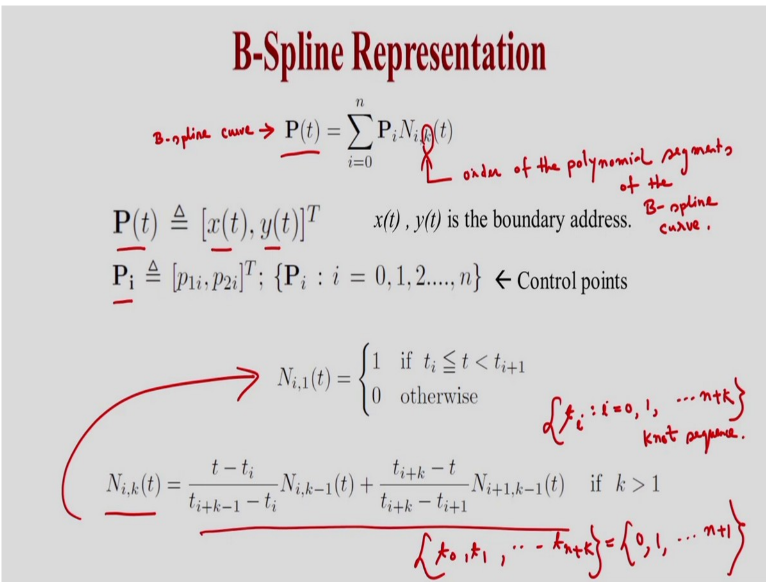

So, this boundary representation by B spline curves. So, the B spline curves are piecewise polynomial function that can provide a local approximation of contours of shapes using small number of parameters. So, that is it is nothing but the B spline representation results in the compression of the boundary data. So, in the figure if you see I have 2 figures, so, figure A and the figure B and in the figure A you can see the B spline curve of degree 3 I am considering and in this case I am considering 8 control points.

So, in the figure you can see I have 8 control points. And the boundary is represented by the B-spline curve and for this I am considering 8 control points. So, these control points divide the B spline curve into number of curved segments and this subdivision of the curve can also be modified. And the B spline curves having a higher degree of freedom can also be considered for curve design.

So, in the right figure if you see, to design a B spline curve, what actually we need, we need a set of control points. So, here you can see I have the control points P0, P1. P2. P3, P4, P5, P6 are the control points and I also need a set of knots. So, here I am showing the knots like this T3, T4, T5, T6 these are knot points and a set of coefficients I need a set of coefficients, one for each control point. And after this, so all the curve segments are joined together satisfying certain continuity condition and the degree of the B spline polynomial can be adjusted to preserve smoothness of the curve to be approximated.

So, this is the concept of the B spline curve. And in the B spline, the B spline allow local control over the shape of a spline curve. So, this approximation I can do by using the B spline curve and this is nothing but the compact representation of a boundary. So, instead of storing all the boundary pixels, I can store only the control points, the knots and also the B spline function. Now, what is the B spline function in the next slide you can see.

(Refer Slide Time: 4:52)



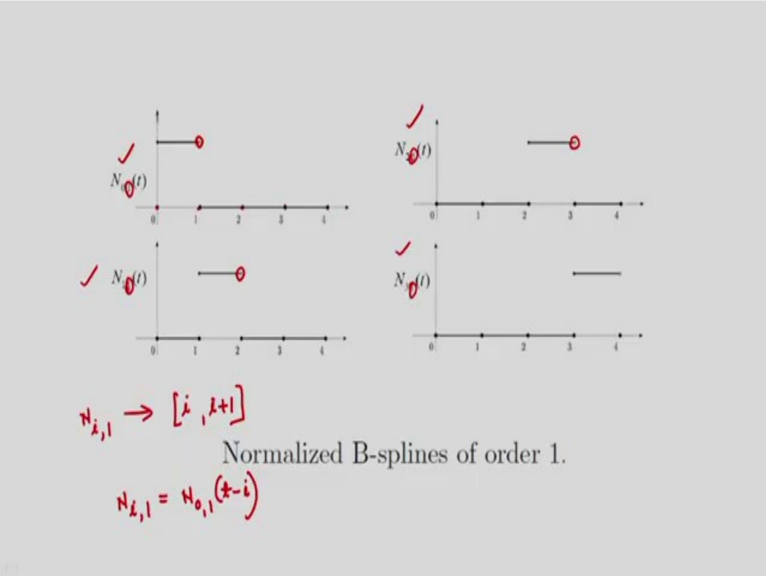
Now, the B spline curve is represented by PT, this is a B spline curve, so it is the B spline curve that is PT is equal to summation i is equal to 0 to N Pi N i k t. Now, in this case the boundary address the because the boundary is nothing but x coordinate and the y coordinate. So, XT and YT and that is represented like this it is represented by P t that is the boundary address. And also I have the control points. So, what are the control points the control points are P i so, control points are P 1 I, P 2 i like this. So, I have the control points.

And in this case the k is the order of the polynomial segment of the B spline curve. So, you can see here what is k, so k is the order of the B spline curve, so I can write it like this. So, the k is the order of the polynomial segments of the B spline curve. So, this is the polynomial segment of the B spline curve. The normalized B spline blending functions of order k is defined by N i,k, N i,k is the the normalize the B spline blending function of order k, that can be defined like this.

And in this case, I need a non decreasing sequence of the real numbers. So, what are the real numbers I can consider, the real numbers are ti and i is equal to 0, 1 up to N plus k, this is ti. Ti is column knot sequence It is called a knot sequence, it is called the knot sequence. So, knots are the locations where the spline functions are tied together. So, I am defining the control points and also the knots and I have shown that B spline of order k. So, this is the B spline of order k.

Now, the B spline of order 1 that I can get here. So, this is the B spline of order 1 and is the knot sequence of a B spline is uniform then it is quite easy to calculate these functions, the B spline functions. So, suppose if I consider the knot t 0, t 1 like this t N k N plus k, I am considering these are the k knot points 0, 1 up to N plus k. So, if the knot sequence of a B spline is uniform, then it is quite easy to calculate these functions this function means the B spline functions. So, in this case I have defined the B spline curve.

(Refer Slide Time: 8:45)

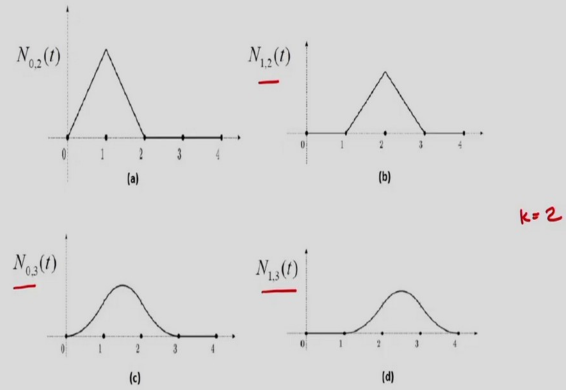



And in this figure first I am showing the B spline of order 1 and after this I have shown N 1 1 after this I have shown N 2 1 and after this I have shown N 3 1. And in this case how many knots I am considering 5 knots in this case you can see 0, 1, 2, 3, 4 So, that means I am considering 5 knots and in this case you can see I am plotting the functions, the functions are N 01, N 1 1, N 2 1, N 3 1 I am plotting, that is a B spline function I am considering. The circle at the end of the line indicates that the function value is 0 at that point.

So, in this case you can see a circle here this is a circle, he circle here at the end of the line that indicates that the function value is 0 at that point. And in this case all these functions have support in an interval, particular interval support means the region where the curve is nonzero. For example, in this case, suppose if I consider the Ni 1 that is a B spline of order 1 has a support on i, i plus 1. And in this case, if I consider what is Ni 1 and Ni 1 is nothing but N 01 and t minus ii.

That means, I am considering the shifting. So, that means from N 01 i can determine Ni 1 by sifting. So, this is about the functions N 01, N 1 1, N2 1 and N 3 1 this is a B spline functions and for this I am considering 5 k knot points. These are the normalized B spline of order 1 because in this case I am considering 1, N 01, N 1 1, N 2 1, N 3 1 I am considering. So, that means, I am considering the B spline of order 1, but you can see the starting point the starting point is 0123.

(Refer Slide Time: 11:03)




(a) Normalized B-splines of order 2 ($N_{0,2}$), (b) Normalized B-splines of order 2 ($N_{1,2}$), (c) Normalized B-splines of order 3 ($N_{0,3}$), and (d) Normalized B-splines of order 3 ($N_{1,3}$).

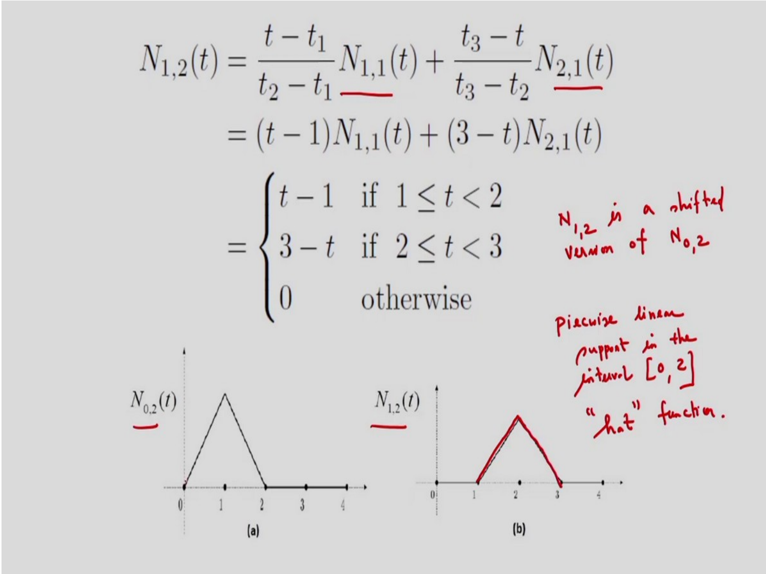
Now, in this case I am showing the normalized B spline of order 2, that means I am considering k is equal to 2. The normalized B spline of order 2 and in this case I am considering N 02 I am considering that is the B spline of order 2 and after this I am considering N 1 2, after this I am considering the B spline of order 3 that is N 03 and after this I am considering N 1 3 that is I am doing the shifting. So, this N 02 can be written as a weighted sum of N 01 and N 1 1.

(Refer Slide Time: 11:40)

$$N_{0,2}(t) = \frac{t - t_0}{t_1 - t_0} N_{0,1}(t) + \frac{t_2 - t_0}{t_2 - t_1} N_{1,1}(t)$$
$$= tN_{0,1}(t) + (2 - t)N_{1,1}(t)$$
$$= \begin{cases} t & \text{if } 0 \le t < 1 \\ 2 - t & \text{if } 1 \le t < 2 \\ 0 & \text{otherwise} \end{cases}$$

So, here you can see in this equation, the N 02 that is the B spline of order 2 can be written as a weighted sum up N 01, the B spline of order 1 and a B spline of order 1 that is the shifted 1 this is about the N 0 2 the B spline of order 2.

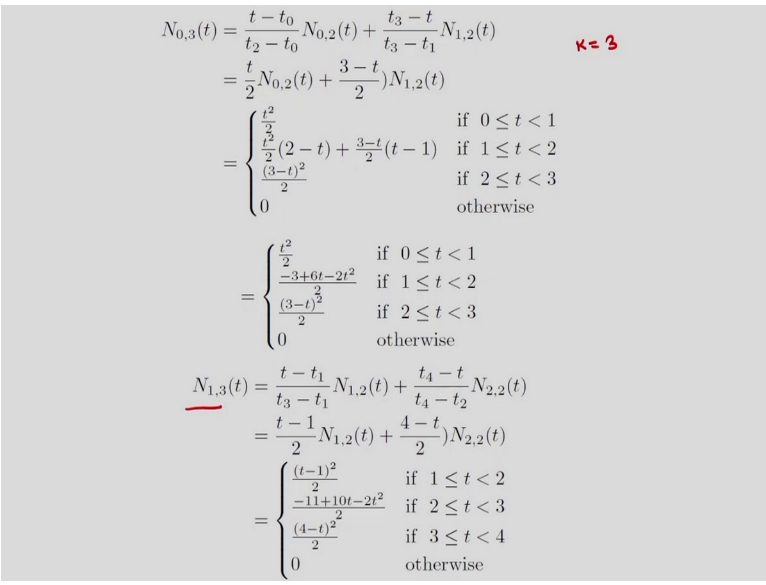
(Refer Slide Time: 12:01)




The next you can see I am considering N 1 2 that I can determine what is N 1 2, N 1 2 I can represent it in terms of N 1 1 and N 2 1 that is the B spline of order 2. So, N 1 2 is a shifted version of N 02. So, I can write this that is N 1 2 is a shifted version of N 02 and in this case you can see I have plotted N 02 and N 1 2. This curve is the piecewise linear curve and the support is from 0 to 2.

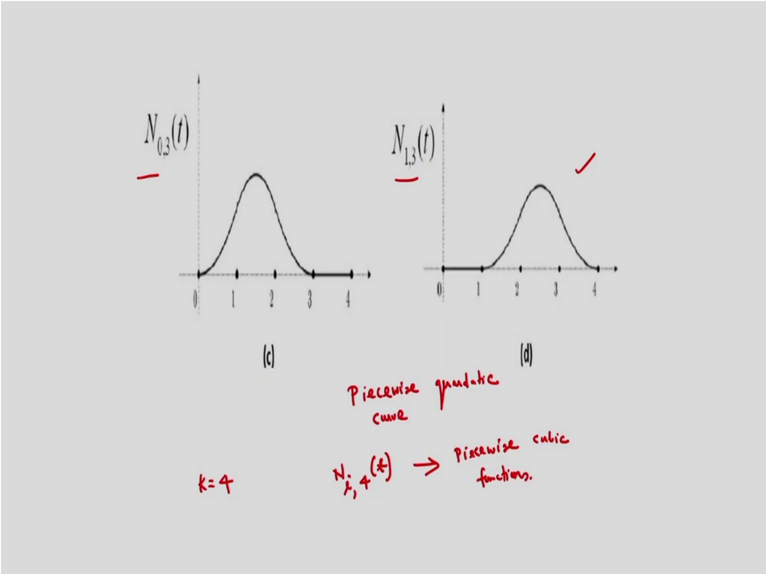
So, this is a piecewise linear and the support in the interval 0, 2 this is commonly known as the head function, this is known as head function. They are used as a blending functions during linear interpolations. So, this if you see this one this is the head function and also I am considering this is the head function.

(Refer Slide Time: 13:43)



And in this case I can show the N 0 3 I can determine that is the N 0 3 is nothing but the B spline of order 3 and N 1 3 that is the shifted version I can determine. Now, if you see these functions N 03 and the N 1 3 in the next slide.

(Refer Slide Time: 14:04)

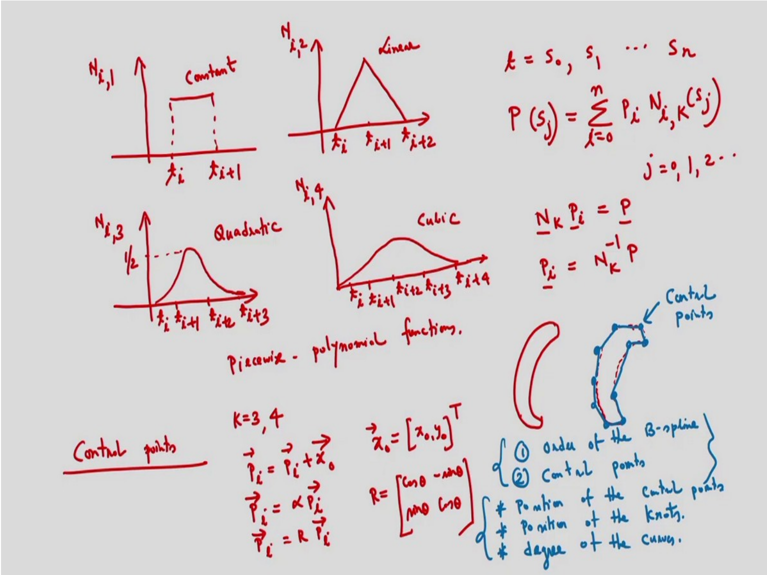



One is the N 0 3 another one is the N 1 3 corresponding to the previous equation these functions are the piecewise quadratic curves. So, I can write like this. So, piecewise quadratic curve. So, these are piecewise quadratic curves. And what is N 1 3? N1 3 is a shifted version of N 03.

So, that is this N 1 3 is nothing but it is a shifted version of N 0 3. And if I consider suppose k is equal to four that is the B spline of order 4, then in this case I will be getting the blending function that is the B spline function and N i, 4 then in this case, I will be getting the the piecewise cubic functions.

So, this is about the B spline functions. So, I have defined the B spline of order 1, B spline of order to B spline of order 3 and a B spline of order 4. So, I can again draw these B spline functions like this, all the B splines I am again drawing.

(Refer Slide Time: 15:36)



So, first one is you can see the B spline, the B spline of order 1, the B spline of order 1 I am considering. For this I am considering 2 knot points ti and ti Plus 1. So, it is constant, that is the B spline of order 1. And if I consider this one the B spline of order 2, so for this I need 3 knot points ti ti plus 1 and t i plus 2. So, already I have defined this is the linear. So, B spline of order 2 and a B spline of order 3 so, what will be the B spline of order 3, this is quadratic.

So, ti ti plus 1, t i plus 2 and t i plus 3 So, this is the B spline of order 3. So, this value is 1 by 2. So, this is quadratic, this is quadratic. And if I considered a B spline of order 4, that is Ni4. So, I need 5 knot points ti plus 1 ti plus 2 ti plus 3 ti plus 4 ti. So, I will be getting. So, I will be getting and the B spline function like this. So, it is called the cubic B spline. So, these are the piecewise polynomial functions, piecewise polynomial functions.

So, generally for computer graphics we consider the B spline of order 3 and the B spline of order 4, the high order B splines. So, this is about the piecewise polynomial functions. And now if I consider the control points. So, how many control points I need for representation of a boundary control points? The number of control points necessary to reproduce a given boundary accurately is usually much less than the number of points needed to trace a smooth curve.

So, that means I need only a few control points to represent a particular boundary. And in this case, I can consider this suppose control point is Pi, this is the suppose vector control points and that can be translated, I can do the translation by a vector. So, what is x0, x0 is nothing but x0 y0 dy. So, I can do the translation, I can do the scaling, I can do the scaling of the control points and also I can do the rotation of the control points.

So, in this case the rotation's transformation already I have explained, the rotation transformation is cos theta minus sine theta, sine theta cos theta, this is the rotation transformation matrix. Suppose we are given the boundary points at the distance values t is equal to suppose S 0, S 1 like this up to SN and corresponding to this we have to find the control points. So, what is the problem, the problem is we are given the boundary points at a distance values of t is equal to S 0, S 1 like this and from this I have to find the control points.

So, I can consider the boundary at this has P Sj and that is summation of i is equal to 0 to n, Pi is the control points and I am considering the Bspline plane of order k in this case j is equal to 0, 1, 2 like this. Now, in this case I can write like this and that is the B spline. This is the B spline of order k and P and P I suppose, N is equal to p. So, from this I can determine the control points Pi is equal to nk inverse p.

So, that means, when the boundary points are given at distance values of T S0 comma s 1 like this, then I can find control points. Now, in this case how to actually represent the boundary. Suppose, if I consider one boundary something like this, this boundary, this is the boundary this boundary is represented by using the control points and by using the B spline functions. So, that means I am approximating the boundaries by considering the control points and the B spline functions.

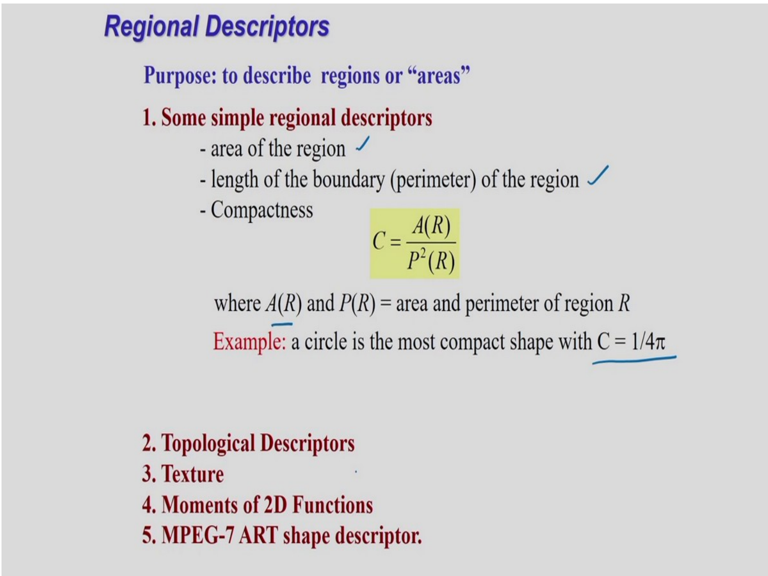
So, I have considering the control points and the B spline functions I am considering. This smoothness I can adjust by considering the order of the B spline. So, if I consider a higher order B spline the smoothness will be more. So, in this case for representing a particular boundary, so what information I need to store in the memory? So, information I need to store in the memories the order of the B spline and the control points.

So, by using these 2 information, I can represent a particular boundary, one is the order of the B spline order of the B spline and next one is the control points. So, that means by using these 2 I can represent a particular boundary. And to change the shape of a B spline curve, one can modify one or more of these control parameters. So, suppose if I want to change the shape of the B spline curve, then in this case what are the parameters I can adjust, one is the position of the control points I can adjust and also I can adjust the position of the knots, that means I can change the position of the knots and also I can change the degree of the B spline curves.

So, that means to change the shape of the B spline curve, I can modify the parameters. So, what are the parameters I can modify, one is the position of the control points, position of the control points, another one is the position of the knots, position of the knots and also I can change the degree of the B spline curve, degree of the curves. So, that means I can change the shape of the B spline curves based on these parameters, one is the position of the control points, the position of the knots and the degree of the curves.

So, in this discussion you can see that to represent a boundary I need to store these 2 information, mainly one is the order of the B spline curve and another one is the control points. And you can see how can I define the B spline curves for order 1, order 2, order 3 and order 4. So, this is about the B spline curve representations.

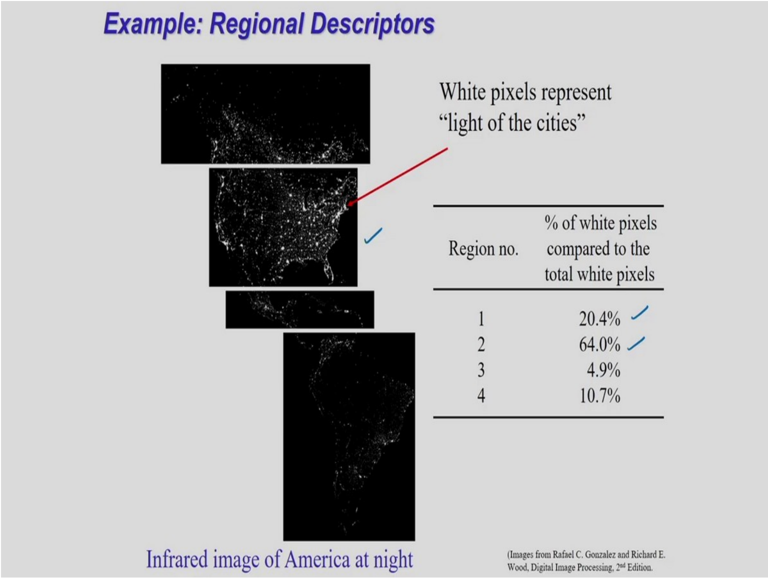
(Refer Slide Time: 24:56)




After this I will discuss the regional descriptors. So, how to do describe a particular region or maybe the particular area. And some of the original descriptors I am going to define here. The first one is I can consider maybe the area of the region or maybe the length of the boundary of the region, I can consider, that is the perimeter I can consider. And also I can determine the compactness. The compactness is nothing but A R divided by P square R. So, what is AR, AR is nothing but the area and PR, the PR is the perimeter of the region lets R.

So, by using these 2 parameters, one is the area another one is the perimeter of the region, I can define the compactness. So, if I consider the circle, corresponding to the circle, the compactness will be C is equal to 1 by 4 pi. So, this is a very simple regional descriptors. After this, the another descriptor is the topological descriptors, the texture descriptors and also the moments of the 2d functions I can determine. And also the MPEG 7 ART shape descriptors. So I am going to discuss about this descriptors, the regional descriptors.

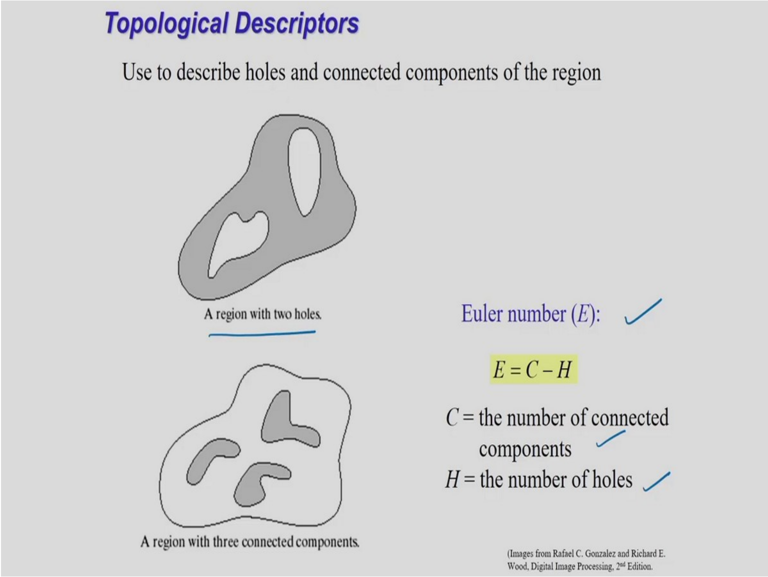
(Refer Slide Time: 26:23)




The first one is you can see the original descriptors. In this case, what I am considering the white pixels represents light of the cities, you can see in the images here. I am considering the input image, that is a binary image and I have the white pixels and I have the black pixels, the black pixels are the background and I have the white pixels that represent light of the cities. And in this case, for regional descriptors what I am considering the percentage of white pixels compared to the total white pixels.

So, for the region 1 it is 20.4 percent, for region 2, 60.4 percent like this I have the percentage. So, by using this concept, I can represent a particular region. This is the most simple regional descriptors.

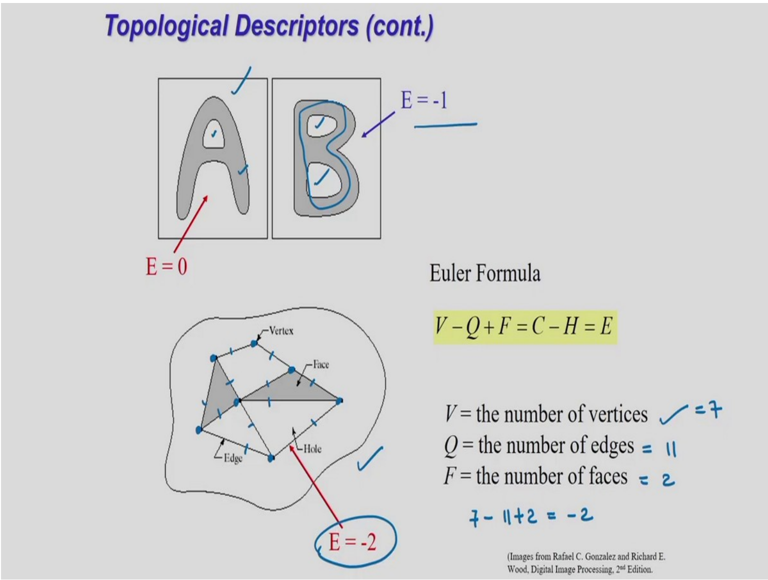
(Refer Slide Time: 27:20)




Another descriptor is that is the topological descriptors. I can consider that Euler number. So, how to define the Euler number? In these 2 figures I have shown in the first figure you can see a region with 2 holes, is the first figure I have shown a region with 2 holes and in the second figure you can see a region with 3 connected components. So, from this you can define that Euler number. So, is the Euler number?

The Euler number is E is equal to C minus H. C means the number of connected components and H means the number of holes. So, by using these 2 parameters, one is the number of connected components and AC the number of holes, we can determine Euler number.

(Refer Slide Time: 28:20)

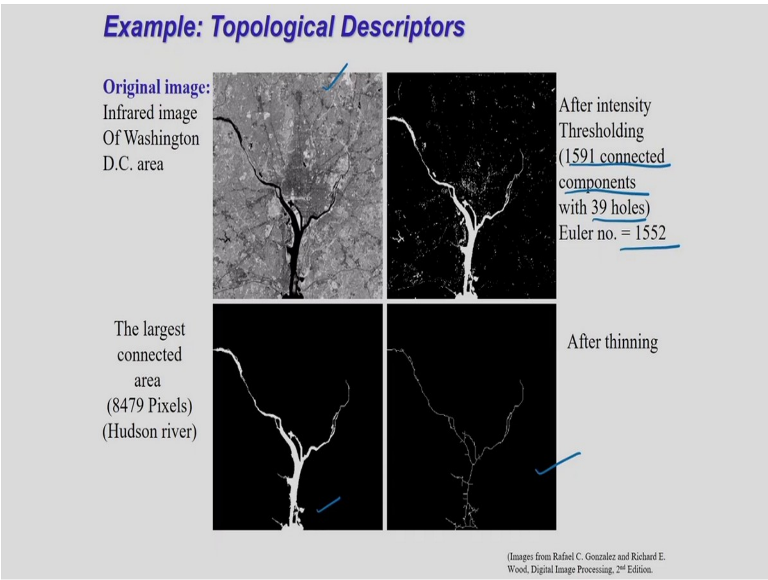


So, in my next slide, I can show how to calculate the Euler number. So, corresponding to this image a the Euler number will be 0. In this case, the number of connected component 1 and only 1 hole, this is the hole and connected component is 1. So 1 minus 1 it is being 0. In the second case, number of connected components, it is 1 because this is the connected component if you see and I have 2 holes, this is 1 hole, another hole. So corresponding to this C minus H that is that Euler number, the Euler number will be minus 1.

This number can be represented also like this, if you see the third figure. So, Euler number you can see V minus Q plus F is equal to C minus H. C means the connected components and H means holes and E is that Euler number. So, what is V, the number of vertices and Q is the number of edges and F is the number of faces. So in this example, if you calculate that number of vertices you can see 1, 2, 3, 4 5, 6, 7 so number of parties is will be 7 in this figure.

And how many edges you can calculate. So edges will be 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. So number of edges will be 11 and number of faces you can see number of faces, I have 2 pieces, so it is 2. So if I put this below here, so V minus Q q is 11 plus F, F is 2. So, I will be getting it is minus 2. So, Euler number will be 2 , E is equal to minus 2 corresponding to this figure, corresponding to this region. So, like this I can calculate the Euler number.

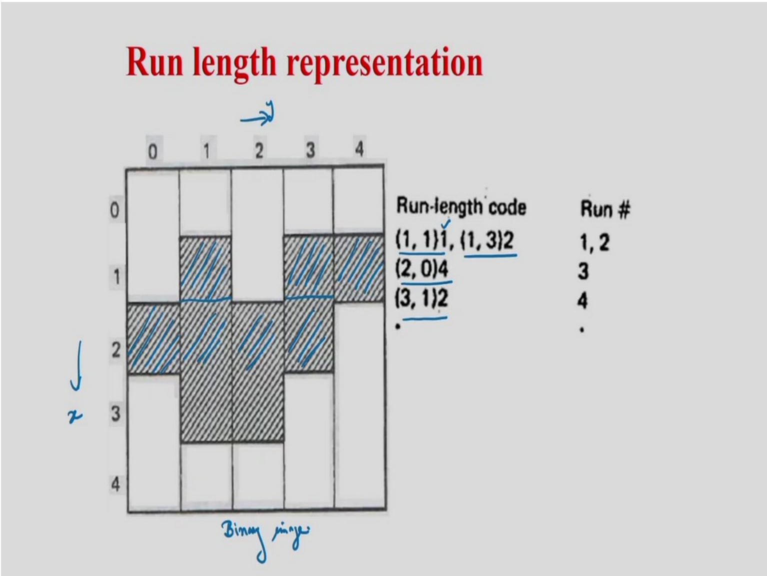
(Refer Slide Time: 30:22)




So, in this example I have shown how to calculate the Euler number and in this case I have shown the intensity image, in this case I am showing the original image, that is the gray scale image and after intensity thresholding I am having 1591 connected components. So, I can calculate how many connected components are there and how many holes are there, thus 39 holes and corresponding to this I can determine that Euler number. So, that means, this image is represented by Euler number.

And in this case, I am considering the largest connected area I am considering, that corresponds to 8479 pixels and after this the morphological thinning I am doing, so I am getting this one. So, that means the main concept is that a particular region is represented by the Euler number.

(Refer Slide Time: 31:19)

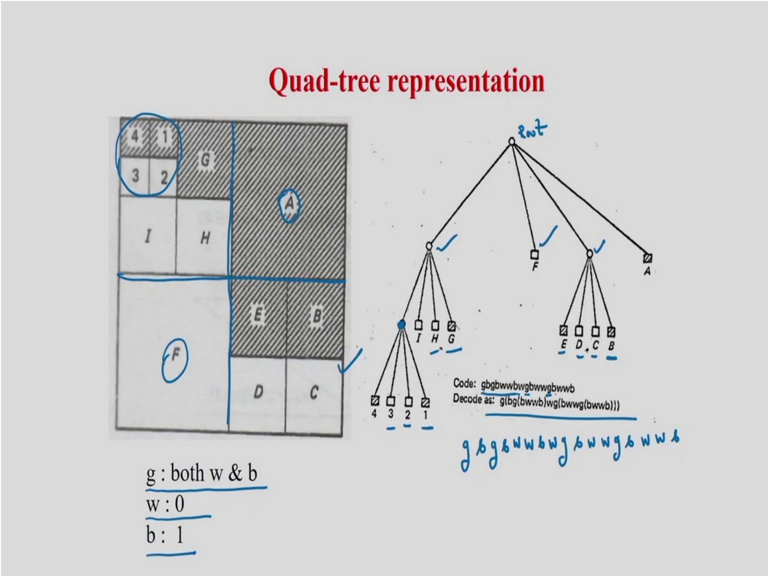



Next, I am going to discuss about the run length presentation of a region. So, in this case I am considering the binary image, so this is the binary image. And you can see suppose if I consider this 1 and 1 pixel that means this pixel. So, how many times it is repeating? So, 1 and 1 pixel is repeating only 1 time, so, it is 1 time is this okay? After this next I am looking for another pixel, the next pixel is 1 and 3, now next pixel this pixel is 1 and 3. This is the x direction if you see this is the x direction and this is the y direction.

So, x coordinate is 1 and y coordinate is 3. So, the next pixel is 1 3, and how many times it is repeated? So, it is repeated twice. So, 1 3 is repeated twice. And if I see the next row, so next row, it is starting from 2 0, the pixel is 2 0, this pixel is to 0 and how many times it is repeating. So, it is repeating 4 times, first, second, third and fourth. So, four times it is repeating. So, that means the pixel 2 0 is repeating four times, and similarly the pixel 3 1 is repeating 2 times.

So, I will be getting the run length code. And by using the run length code, I can represent that image, that is the binary image and this is the run length representation.

(Refer Slide Time: 32:56)




Next I am considering the quad tree. So, by using the quad tree also I can represent a particular region. This concept I have already discussed in my class of image segmentation and that is the concept is split and merge technique. In this case, if the region is not homogeneous, that means I am considering one image if the image is not homogeneous, if the region is not homogeneous, I have to divide it into 4 regions.

So, here you can see the region is not homogeneous, so that is why I am dividing into 4 regions like this. And this is my root node, this is my root. And in this case, if it is homogeneous, then in this case no need to do the splitting, I can merge as per the image segmentation procedure, procedure is split and the merge technique. And if I see the region A that is homogeneous, so no need for further splitting, but if I consider this region, it is not homogeneous. So, that is why I am dividing it into four regions, hat is the splitting it into four regions.

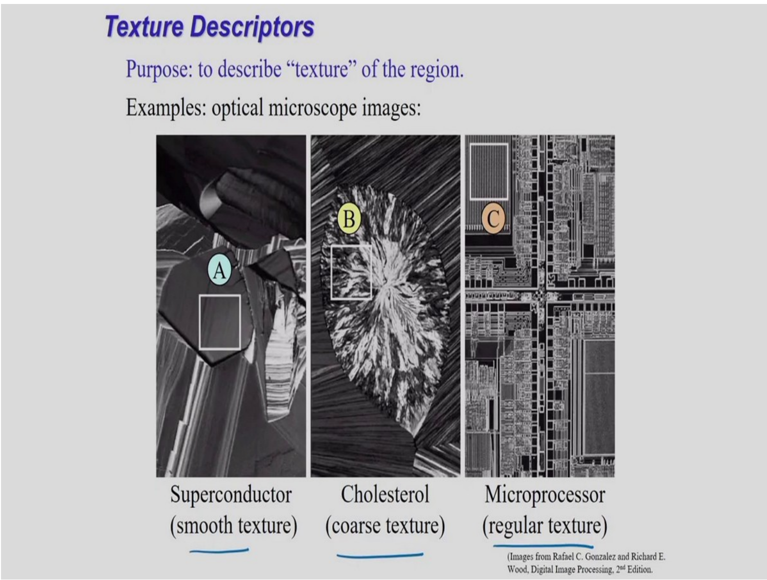
And if I consider this region, this is homogeneous, so that is why no need to do the splitting. And similarly, I am doing this one and you can see this region is not homogeneous. So, that is why I am dividing it into four regions. So that means I am doing the splitting. And based on this I can represent this in the quad tree. So, the root is the image and you can see here I am considering the code here. The code will be something like this. If I considered G, G means the gray, that means I am considering white and the black.

If I consider white, white means 0 and if I consider black the black is 1. So, original image is the gray image because it contains white and black. So, that is why the original image I am writing it is gray, G. After this, I am splitting this one, splitting the original image. So, I am getting the regions. The region is A and that is why the region is the black, after this I am getting the second region, that is it has both black and white. So, that is why I am considering it as gray.

And after this within this gray, you can see I have the black 1. So, this is my black 1 white 1 another white and after this again black. So, up to this I am getting this one. After this if I consider this region this is completely white, so I am considering it as white, this is white. After this if I see this node that is the gray because it contains both black and white, so that is the gray. So, I have the gray this one and in this case, you can see the first I have the black.

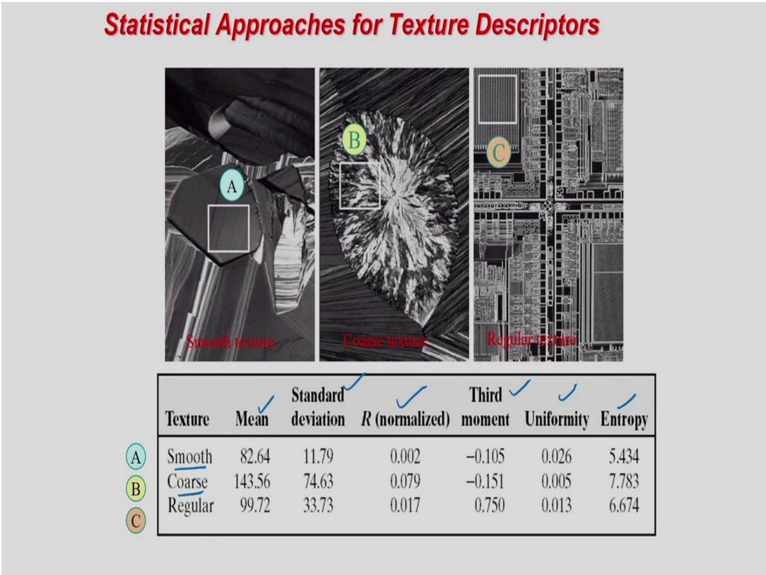
So, this is black after this the white after this the white and if I see this node, it contains both black and white, so that is why I am writing as gray. So, this is a gray and in this gray node, I have the black 1, the black is this after this the white, after this white and after this black. So, like this I will be getting a code corresponding to this image, that is the region corresponding to this region I am getting the code. So, this is called the quad tree representation. So, by using the quad tree representation, I can represent a particular region.

(Refer Slide Time: 37:01)



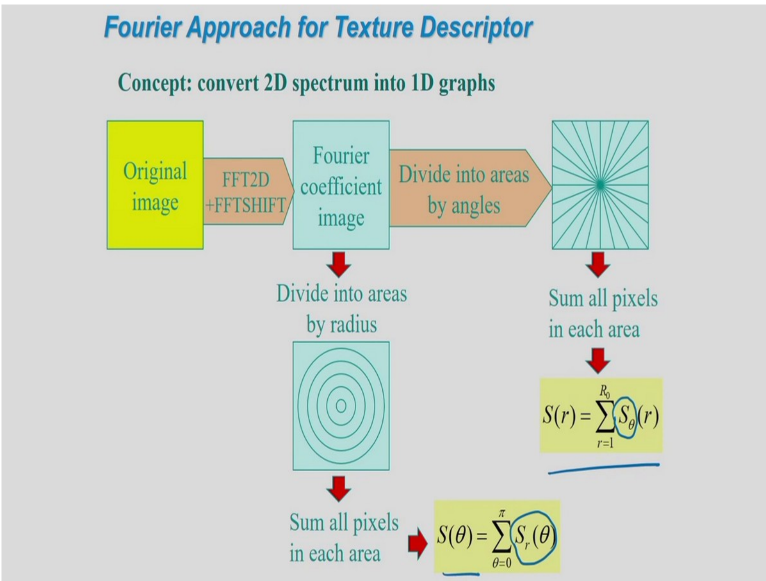

And already I have discussed about the texture descriptors. So, I am not going to discuss it again. And you can see I have considered the smooth textures, the coarse textures and the regular textures. And for this I can determine some statistical parameters I have already discussed about this and also I can determine GLCM the gray level co occurrence matrix and from this I can determine some quantities to represent the textures.

(Refer Slide Time: 37:34)



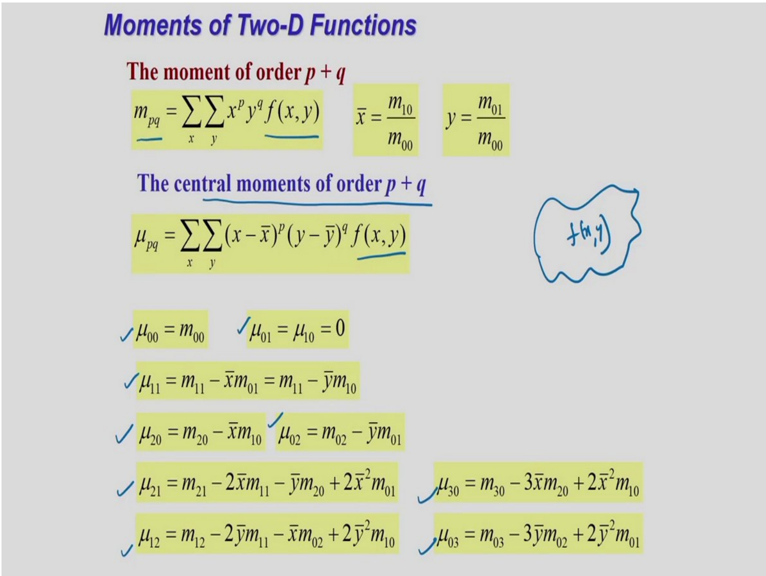

So, you can see I can determine the statistical parameters like the mean, standard deviation, roughness factor, third moment, uniformity, entropy, and by using this I can represent different types of textures, that I can represent the smooth textures, the coarse textures and the regular textures.

(Refer Slide Time: 37:55)




And also I discussed about the Fourier approach for texture descriptors. And that concept also I have discussed. So, for this I have to determine the Fourier transform of the image and after this I have to convert the Fourier transform into polar representations. And after this I have to determine these 2 spectrums, one is SR another one is S Theta. So, that means in case of the this one S theta, the theta is fixed and R is variable. And in the in this case the R is fixed the theta is variable. So, I will be getting these 2 spectrums, one is s r Another 1 is S theta and by using this I can represent a particular texture.

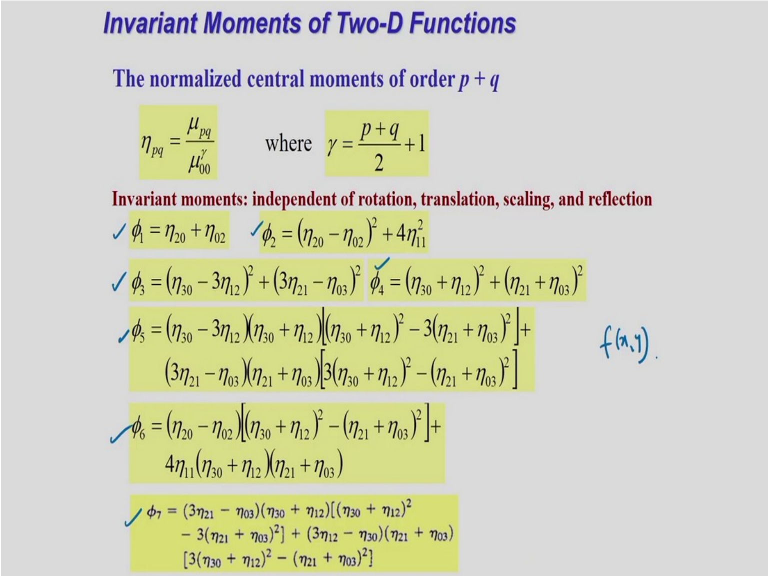
(Refer Slide Time: 38:40)




So, by using the image moments, I can represent a particular region. The image moments corresponding to the image Fx y, I can determine like this mpq, I can determine okay and also I can determine the central moments of order p plus q. So, I can determine the central moment of order p plus q for the image F xy by using this expression and from this I can determine the moments mu 00, mu 01, mu 1 1, mu 2 0, mu 02, mu 2 1, mu 1 2 mu 3 0, mu 03 I can determine.

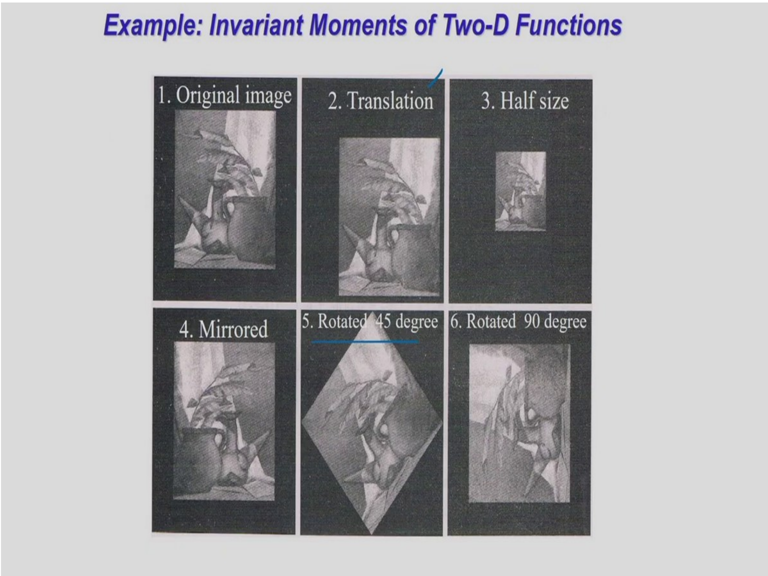
So, that means corresponding to a particular region, that means suppose corresponding to this region suppose it is Fx y. So, corresponding to this Fx y by can determine this moment I can compute the central moment of order p plus q.

(Refer Slide Time: 39:33)



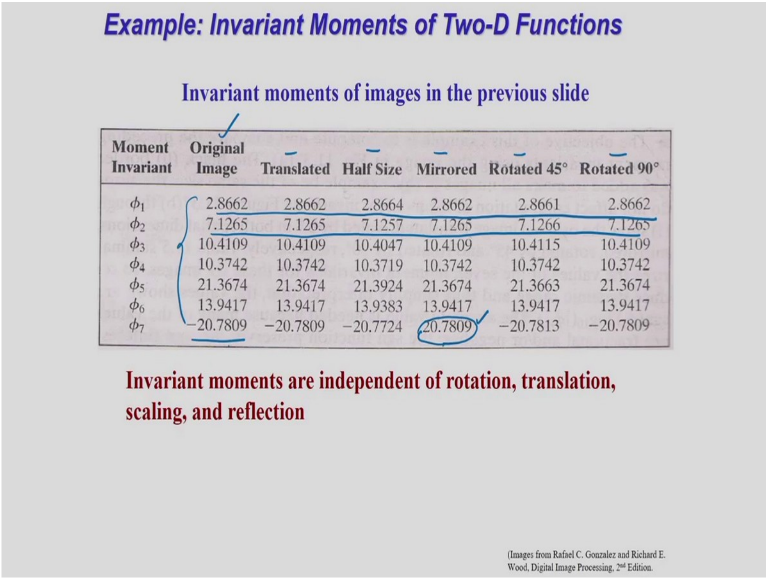

And I can determine invariant moments, that is the invariant moment means it is invariant to a fine transformation. So, I can determine 7 moment invariance. So I can determine phi 1, phi 2, phi 3, phi 4, phi 5, phi 6, phi 7 I can determine 7 invariant moments which are independent of rotation, translation, scaling and reflection, so that I can determine. So, from the image I can determine this the image is F xy. So, from the image I can determine 7 invariant moments.

(Refer Slide Time: 40:16)




And in this case, I have shown one example, you can see the original image, the second one is the translated image. The third one is the scale image, that is the half size image, the fourth one is the mirrored image, you can see the mirrored image, that is the 180 degree rotation, the fifth one is rotated by an angle of 45 degree, rotation by an angle of 45 degree and the number 6th is rotation by an angle of 90 degree rotated 90 degree.

(Refer Slide Time: 40:50)

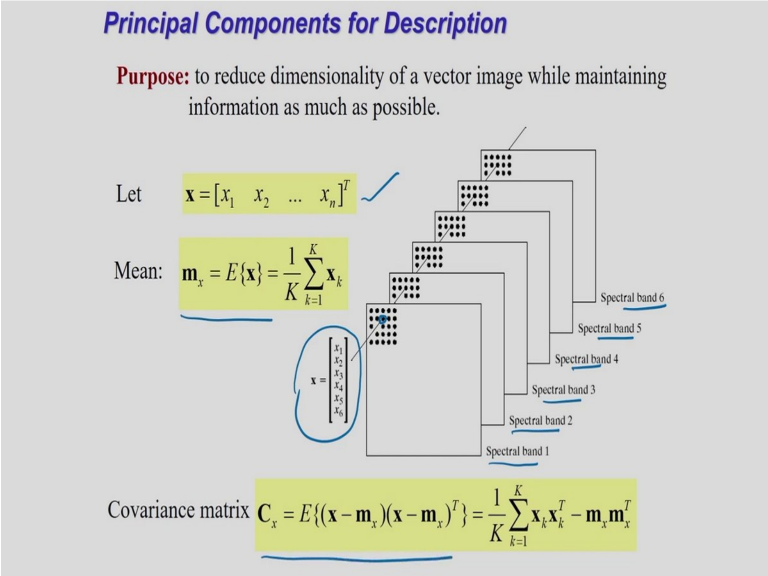



| Moment Invariant | Original Image | Translated | Half Size | Mirrored | Rotated 45° | Rotated 90° |
|---|---|---|---|---|---|---|
| $\phi_1$ | 2.8662 | 2.8662 | 2.8664 | 2.8662 | 2.8661 | 2.8662 |
| $\phi_2$ | 7.1265 | 7.1265 | 7.1257 | 7.1265 | 7.1266 | 7.1265 |
| $\phi_3$ | 10.4109 | 10.4109 | 10.4047 | 10.4109 | 10.4115 | 10.4109 |
| $\phi_4$ | 10.3742 | 10.3742 | 10.3719 | 10.3742 | 10.3742 | 10.3742 |
| $\phi_5$ | 21.3674 | 21.3674 | 21.3924 | 21.3674 | 21.3663 | 21.3674 |
| $\phi_6$ | 13.9417 | 13.9417 | 13.9383 | 13.9417 | 13.9417 | 13.9417 |
| $\phi_7$ | −20.7809 | −20.7809 | −20.7724 | 20.7809 | −20.7813 | −20.7809 |

So, corresponding to this I can determine the moment invariance, these are the 7 moment ingredients, the phi 1, phi 2, phi 3, phi 4, phi 5, phi 6, pi 7, I can determine and you can see corresponding to all these cases, this is my original image, corresponding to the original image I have these values corresponding to phi 1, phi 2, phi 3 up to phi 7 and corresponding to these cases the translated image, the scale image, the mirrored image rotated image rotated image, I can determine these values the phi 1 up to phi 7. And in this case, corresponding to these rotations and all these fine transformations you can see this value is almost same, that is it is invariant to a fine transformation.

Similarly, if I consider the second moment phi 2, that is also almost constant for all these cases. Except for a mirrored image this value is positive, but here this value is negative. Except for the mirror image and this value is positive and this value is negative. So, from this you can see that I have the 7 invariant moments for representing a particular image. These invariant moments are independent of rotation, translation, scaling and reflection.

(Refer Slide Time: 42:17)

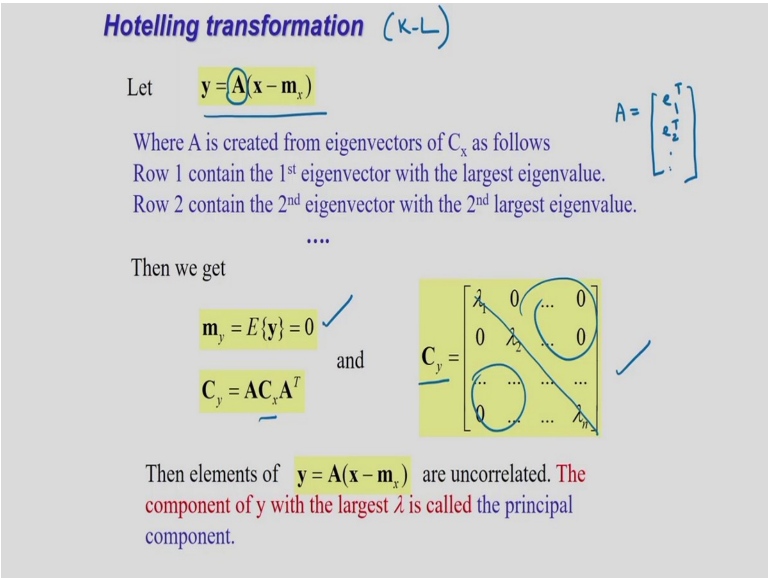



After this, I will just briefly discuss about the principal component for image description. So, in my image transformation class I have discussed about the KL transformation, I discussed about the PCA, the principal component analysis, the same concept is used for representing images, that is the principal component for image description. And in this case, the purpose is to reduce the dimensionality of a vector image while maintaining information as much as possible.

So, in this case, in this example, I am considering different images captured in different spectral bands. So, spectral band 1, spectral band 2, spectral band 3 like this, so, different spectral bands I am considering. And if I consider a pixel suppose, these pixels from all the spectral bands I am considering and that can be considered as a vector, the vector is x.

Suppose one pixel I am considering and I am considering all the spectral bands, in this example, I am considering the 6 spectral bands. So, I am having done this the vector, vector is x, x is the input vector. From this input vector I can determine the mean of this vector and also I can determine the covariance matrix I can determine. That this concept already I have explained in KL transform.

(Refer Slide Time: 43:51)




After this I am considering the Hotelling transformation, Hotelling transformation is nothing but that KL transformation and this is also the KL transformation. So, transformations is y is equal to a x minus mx that is the KL transformation, that is the Hotelling transformation. In this case a is the transformation matrix. So, how to construct that transformation matrix? So, the matrix A is constructed from the Eigen vectors of the covariance matrix.

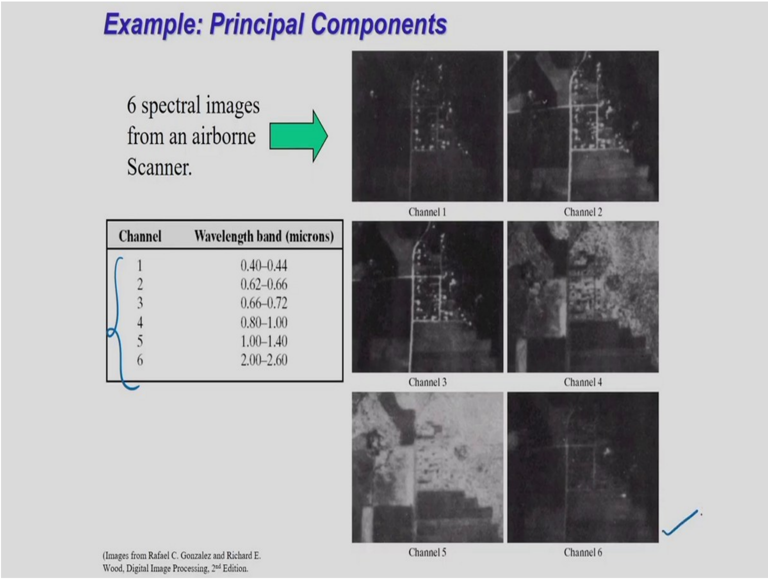
The matrix A that is a transformation matrix is constructed from the Eigen vectors of the covariance matrix. So, the row number 1 contains the first Eigen vector with the largest Eigen value. So, this is the first row of the transformation matrix. What about the second row of the transformation matrix? Row 2 contains the second Eigen vector with the second largest Eigen value. So, like this I can determine transformation matrix.

So, like this e1 T, first Eigen vector, the second Eigen vectors, like this I can determine the transformation matrix. After the transformation what is happening, you can see the mean of y that is the mean of the transformed data will be 0. And I can determine the covariance matrix in terms of the covariance matrix of x, x is the input data, the covariance matrix of y, y is the transformed data, I can determine the covariance matrix of the test data in terms of the covariance matrix of x, that is the Cx.

And you can see, I have the covariance matrix of y of the transformed data and I have the diagonal covariance matrix and these off diagonal elements are 0 all are 0. That means, the transformed data will be uncorrelated. So, that means, the elements of y are uncorrelated. That is the transformed data will be uncorrelated the component of y with the largest lambda is called the principal component. So, if I considered a component of y we the largest lambda lambda is the eigenvalue is called a principal component.

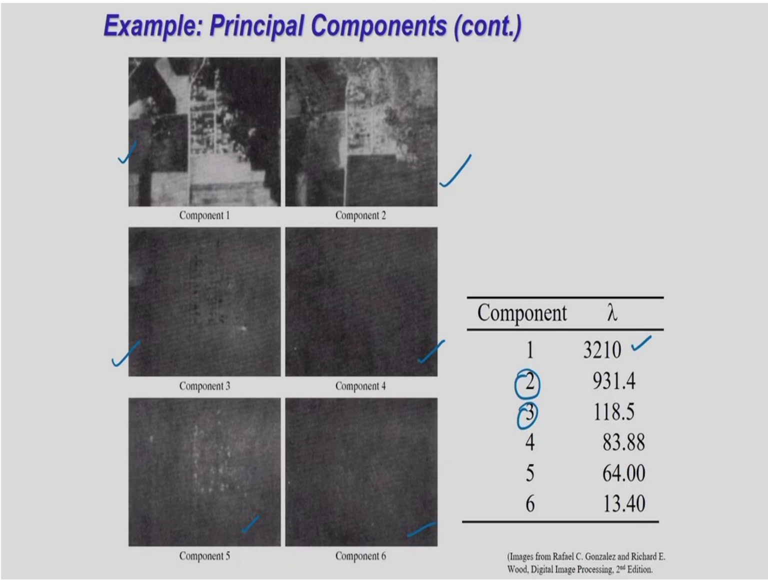
So, by using this method, the Hotelling transformation or the KL transformation or I can consider PCA the principal component analysis, how to reposition an image.

(Refer Slide Time: 46:19)




Here I am considering 6 images corresponding to 6 spectral bands. So, I am considering the 6 spectral bands, Channel number 1, Channel number two, channels number 6 up to. So, you can see I have the images corresponding to all the spectral bands channel 1, Channel 2, Channel 3, Channel 4, Channel 5, Channel 6 so, I have 6 images.

(Refer Slide Time: 46:49)

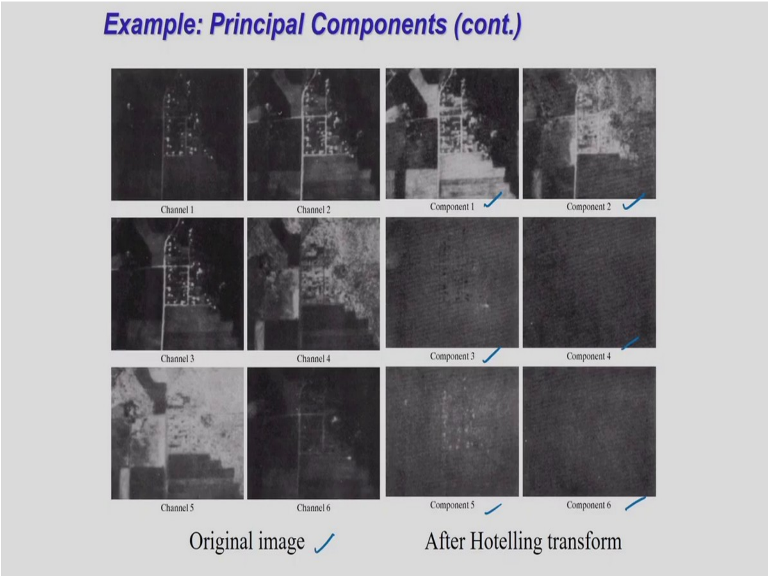


And from this I can apply the Hotelling transformation, that is the KL transformation. And in this case, I am considering the principal component you can see corresponding to the largest Eigen value the number 1 and you can see the Eigen values like this 3210, 931, 118 like this and these are represented as an image. So, if I only considered the first eigen value, that is the principal component corresponding to this, this will be my image, the component 1. If I considered a component 2, that is this component that will be my image.

If I consider a component 3 corresponding to the eigenvalue that will be my image component 4 is like this, component 5, component 6. So, in this case, the component 1 corresponding to the largest Eigen value have maximum visual information as compared to component 3, component 4, component 5, component 6.

So, for the presentation of the images, I can only select component 1 and component 1 and the component 2, I can discard component 3, component 4, component 5, component 6. So, that means by using only 2 principal components, I can represent the image that is nothing but the dimensionality reduction.

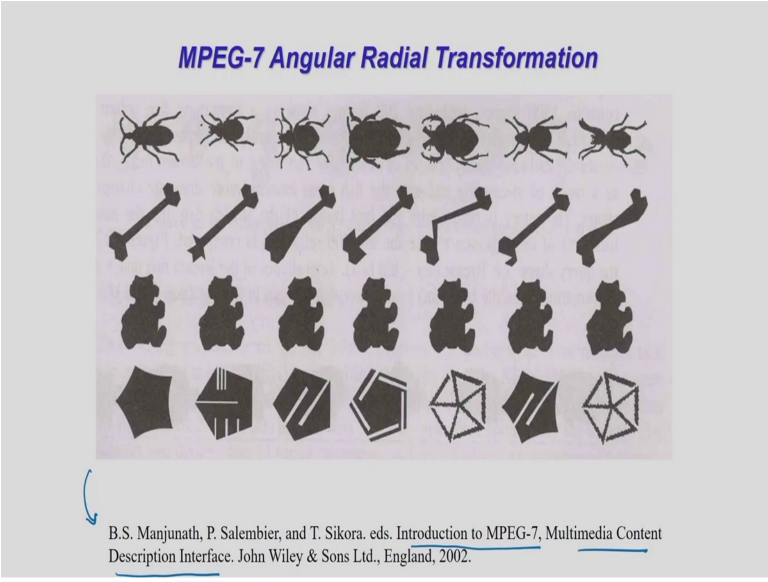
(Refer Slide Time: 48:09)




So, here I have shown the original image Channel number 1, Channel number 2, Channel number 3, Channel number 4, Channel number of 5, Channel number 6 I have the original image and I have shown I have shown that components these are the first one is the principal component, the component 1, component 2, component 3, component 4, component 5, component 6 after the KL transformation. And already I have explained the component 1 and the component 2 contain maximum visual information.

So, I can consider only these 2 components, component 1 and component 2 and I can neglect component 3, component 4, component 5 and component 6. This is the method to represent a particular image by using and Hotelling transformation.

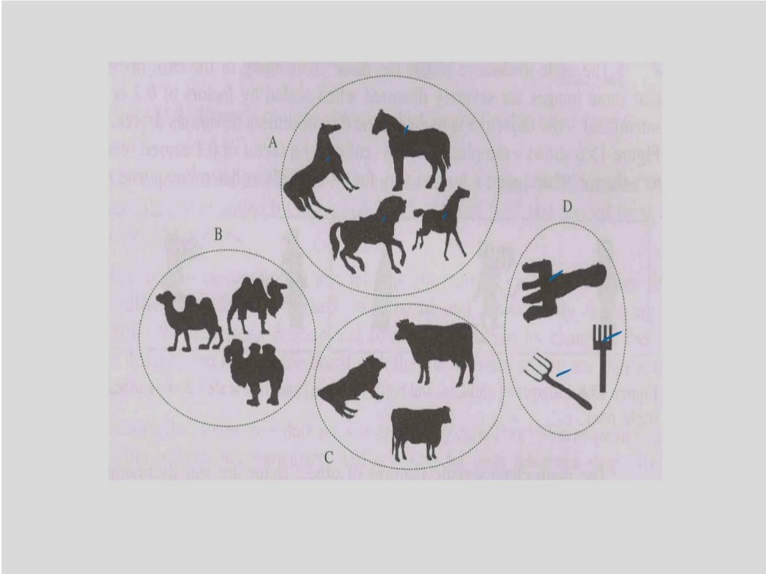
(Refer Slide Time: 48:58)




And finally, I want to discuss their MPEG 7 angular radial transformation. This MPEG 7 is called the multimedia content description interface. This is not used as an compression standard. I am considering some binary images. So, how to represent these binary images by using MPEG 7 Angular radial transformation.

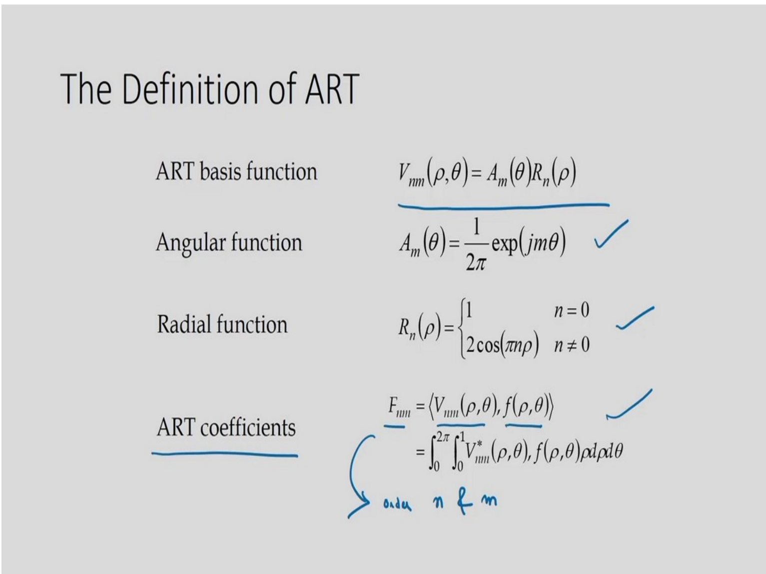
That is the shape representation by MPEG 7 RT, ART means the angular radial transformation descriptors. And for details you can see the book by Professor Manjunath, that introduction to MPEG 7, multimedia content description interface. So, in my class only I am giving the highlight of this technique, the MPEG 7 ART shape descriptors.

(Refer Slide Time: 49:52)




So, in this case based on the shape similarity, I can do the clustering. So, in this case, so if I consider these shapes, these are almost similar, so this will be one cluster. And similarly, if I consider these shapes these are almost similar. So, I can do the clustering like this based on the shape. And also for content based image retrieval, I can use this technique. So, I will explain how we can use the MPEG 7 ART shape descriptors for shape retrieval, the image retrieval.

(Refer Slide Time: 50:25)

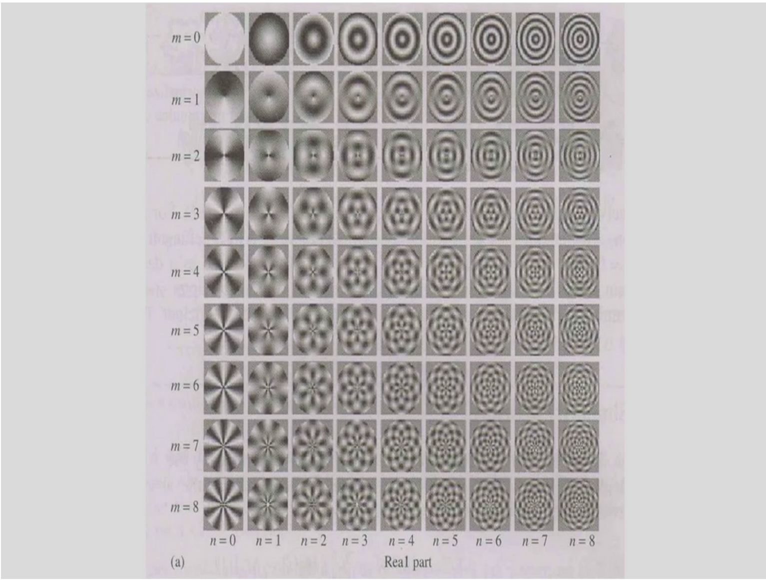



So, what is the definition of the MPEG 7 ART descriptors. So, here you see this is the ART coefficients I can get by using this expression. So, in this case F rho Theta that is nothing but the image the image function that is represented in the polar coordinate. And V nm is the

ART basis function. So, my basis function is V nm. So, that is I am considering the inner product between Vnm that is the ART this is function and the image the image is F rho theta. So, ART is the orthonormal unitary transformation defined on a unit dx that consist of the complete orthonormal sinusoidal basis functions in polar coordinates.

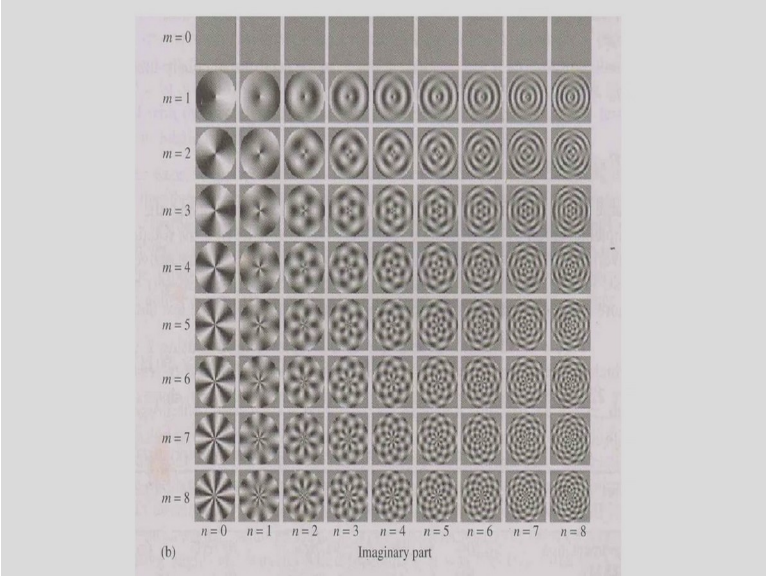
So, in this case Fnm, Fnm is the ART coefficients, ART coefficients that is the angular radial transformation coefficients of order, order is given the order is and n and m. That is the ART coefficients here the coefficients of order n and m. And this ART basis function is separable along the angular and the radial direction. So, we have the ART basis function that is separable along the angular and the radial direction. So, that is the angular function and this is the radial function corresponding to the ART basis function. And from this you can easily calculate the LT coefficients.

(Refer Slide Time: 52:04)



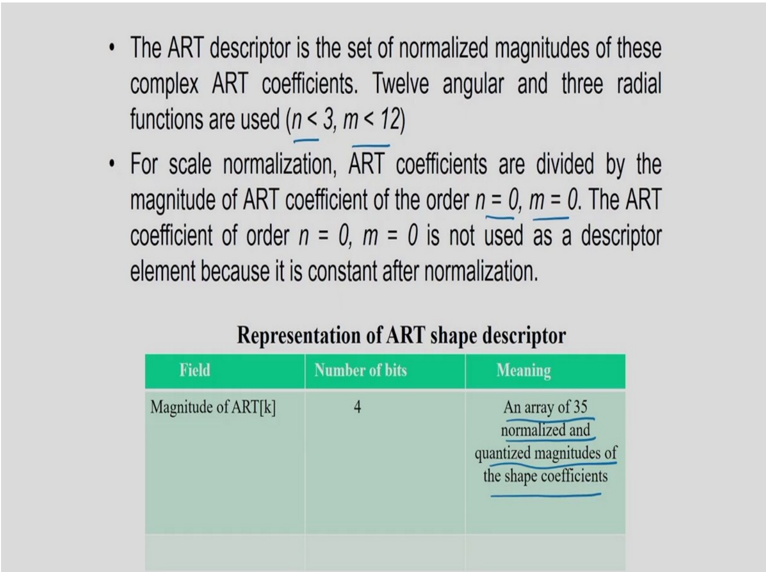

And in this case, I have shown the real part of the basis function, the ART basis function you can see, I am showing the coefficients that n0, n1, n2 these are the I am showing n8 and here also I am showing m0, m1 like this, these are the real parts of the basis function ART basis function.

(Refer Slide Time: 52:24)




And similarly, I have the imaginary part corresponding to the basis function. This by using the previous expression that you can determine, this real part and the imaginary part you can determine from the previous expressions.

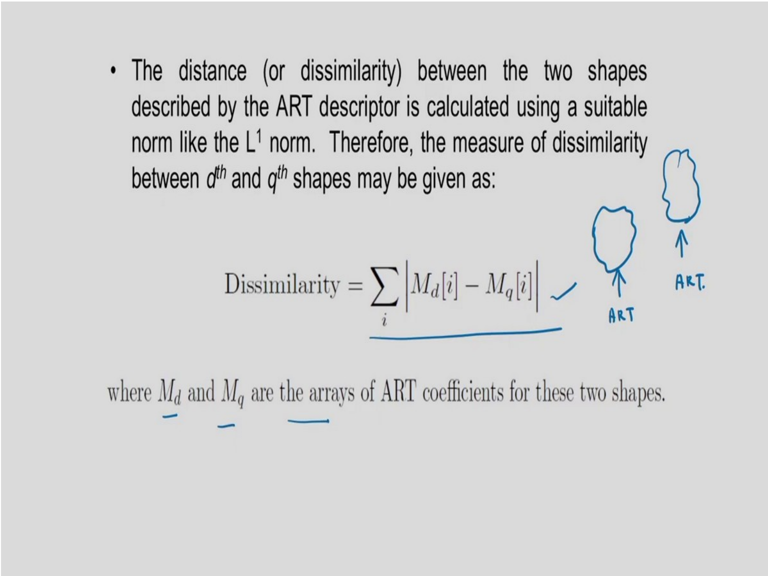
(Refer Slide Time: 52:41)




And this the ART descriptor is a set of normalized magnitude of these complex ART coefficients and the 12 Angular and the 3 radial functions are used. So, that means, n is less than 3 and m is less than 12 I am considering and corresponding to this if I neglect this one the ART coefficients of order n is equal to 0 and m is equal to 0 that is used for normalization.

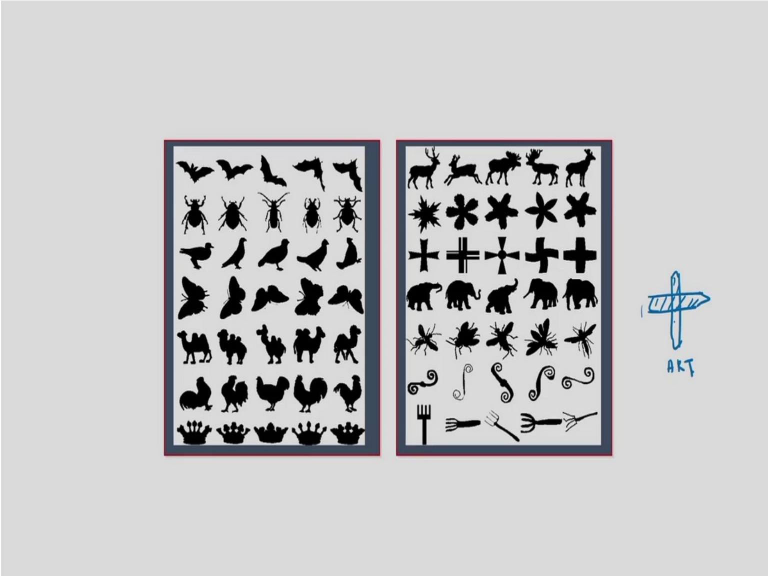
So, if I neglect this, then in this case I will be having only 35 normalized ART coefficients. So, that means, I will be getting an array of 35 normalized and quantized magnitude of the ART coefficients, because this value I am not considering the ART coefficients corresponding n N is equal to 0 and m is equal to 0 that is not considered, that is used for that normalization.

(Refer Slide Time: 53:34)




And in this case, if I want to consider the similarity between suppose 2 shapes, suppose one shape is this and another shape is this. So the dissimilarity between these 2 shapes can be determined based on the ART coefficients. So, I am calculating dissimilarity between 2 shapes, then in this case, you can see Md and the Mq are the arrays of the ART coefficients for these 2 shapes. For this shape also we have to determine the ART coefficients and for these also I have to determine the ART coefficients and after this, I can compare these 2 shapes by using MPEG 7 ART shape descriptors. So, I can find the dissimilarity.

(Refer Slide Time: 54:21)

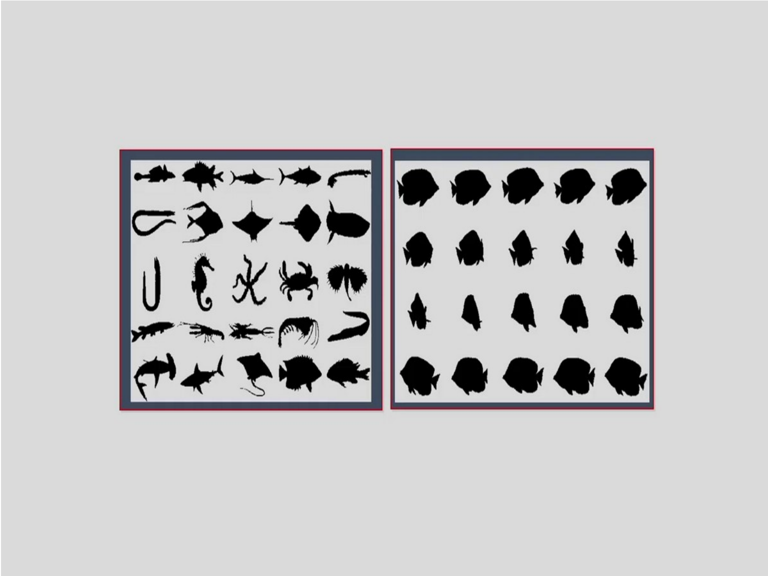



And one application already I have mentioned, that is the content based image retrieval. Suppose, in my database, I have these all the images. And suppose, if I want to retrieve a particular image, suppose if I want to retrieve this particular image from the database, then what I have to consider corresponding to this shape, I have to determine the ART coefficients and for all other shapes I am storing the ART coefficients corresponding to a particular shape.

So, we need to store the image, we have to store the ART coefficients corresponding to a particular shape. And after this for comparison what I have to consider I have to consider the ART coefficients corresponding to 2 steps. And based on the dissimilarity measure, I can select a particular image from the database, that is the content based image retrieval by considering MPEG 7 ART shape descriptors.

So here you can see, so corresponding to all the images here, I can determine ART shape descriptors, ART coefficients I can determine. And corresponding the query image, suppose this is my query means, I can also determine the ART coefficients. After this I have to find out the similarity between the shapes based on the ART coefficients and based on this I can retrieve a particular shape from the database, that is the content based image retrieval.

(Refer Slide Time: 55:51)



And similarly, I can show some of the images like this, the binary images and corresponding to all the binary images, I can determine the ART shape descriptors. So, I need not store the image, only I have to store 35 ART coefficients because each shape is represented by 35 coefficients and which are ingredient to a fine transformation. So, this is about the MPEG 7 ART shape descriptors.

In this class I discussed 2 important descriptors, one is the ART shape descriptors that is the MPEG 7 ART shape descriptors. And also I discussed about the B spline representation of a curve. So, in the B spline representation of the curve, to represent a particular curve or to represent a particular boundary, I have to consider the B spline curve and for this I have to store the control points and the order of the B spline. By considering this information I can represent a particular boundary. This is about the B spline.

And regarding this MPEG 7, that is called a multimedia content description interface, I can represent a particular shape by using ART, that is the angular radial transformation shape descriptors. This is about the shape and the boundary representation by descriptors. So, I can consider boundary descriptors, I can consider the region based descriptors. So, let me stop here today. Thank you.