Digital Image Processing

Prof. P. K. Biswas

Department of Electronics and Electrical Communication Engineering

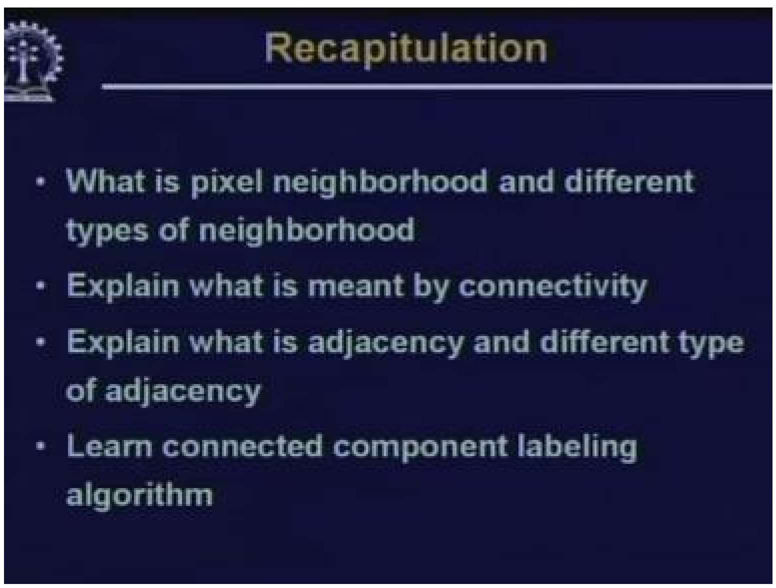
Indian Institute of Technology, Kharagpur

Lecture - 5

Pixel Relationships II

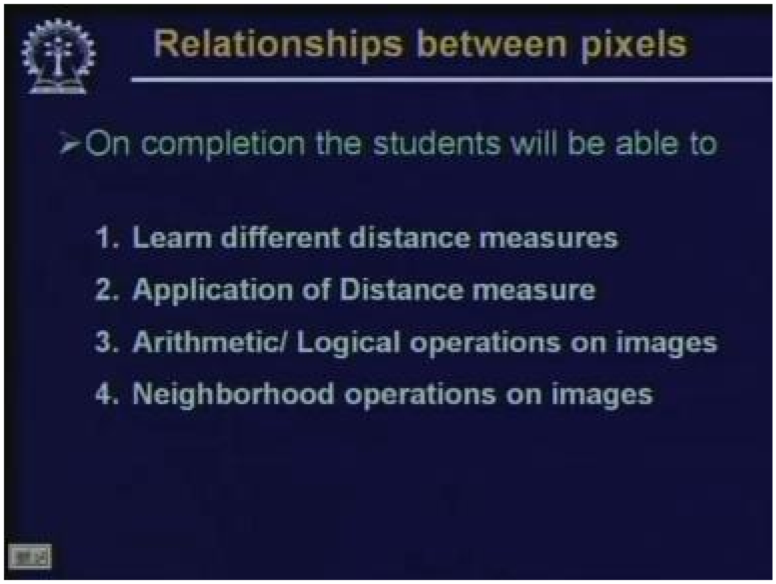
Hello, welcome to the video lecture series on digital image processing. In our last class, we have started our discussion on pixel relationships. Today we will continue with the same topic of discussion on pixel relationships. In the last class, we have seen what is meant by pixel neighborhood and we have also seen different types of neighborhood.

(Refer Slide Time: 1:29)



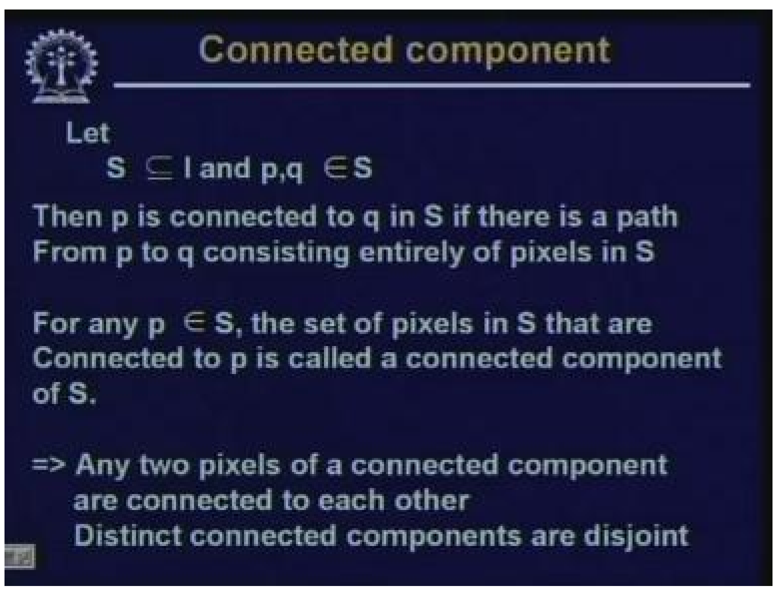
We have explained what is meant by connectivity, we have seen what is adjacency and different types of adjacency and we have also seen a connected component labeling problem.

(Refer Slide Time: 1:46)



In today's lecture, we will learn different distance measures, we will see application of distance measures, we will see arithmetic and logical operations on images and we will see neighborhood operation on images.

(Refer Slide Time: 2:07)



So, let us first repeat what we have done in the last class that is on connected component labeling. Connected component leveling is a very very important step specifically for high level image understanding purpose. Because, as we have seen earlier that if an image contains a number of objects, then the image region can be divided into 2 regions - object region and background region. And there you have seen that you have taken a decision role that if the intensity value at a particular region is above certain threshold, then we assume that particular pixel belongs to the object and if the intensity is less than the threshold, we assume that the particular pixel belongs to the background.

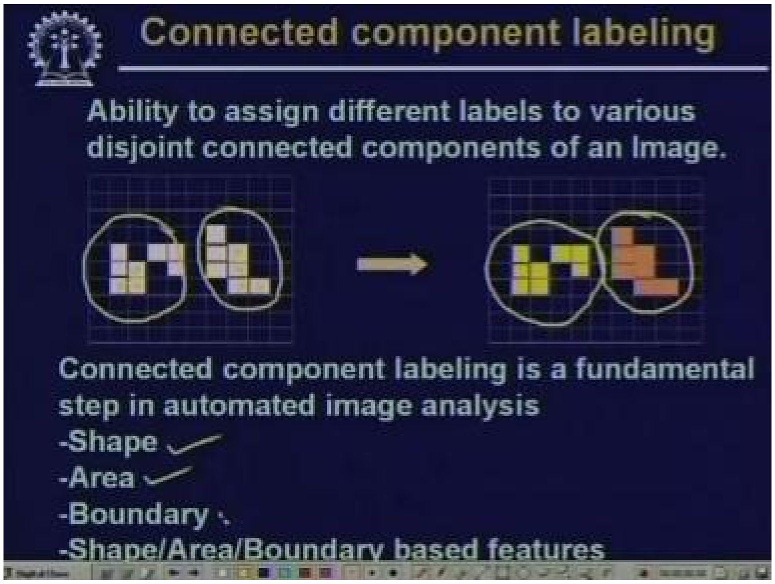
Now, this simple rule, simply tells that a pixel whether it belongs to an object or it belongs to a background. But if there are multiple number of objects present in the same, this simple rule does not tell which pixel belongs to which particular object where the component leveling algorithm or the component labeling problem gives a solution because it keeps tells you not only the pixel belonging to a particular object but it also associates every pixel to a particular object.

So, once you have this association of the pixels belonging to a particular object, you can find out different properties of the particular object. For example; what is the area of that object region, what is the shape of the object region, what is the boundary of that object region and different other shape related features which can be used for a high level understanding purpose.

Now, let us see that what is meant by a connected component. Here we assume S to be a subset of an image I and let us assume that we have 2 pixels p and q which belongs to the image subset S. Then we say that the pixel p is connected to pixel q or point p is connected to point q in the image subset S if there is a path from p to q consisting entirely of pixels in the region S.

So for any point p belonging to S, the set of pixels in S that are connected to p is called a connected component of p. So, any 2 pixels of a connected component are connected to each other and you find that distinct connected components have to be disjoint.

(Refer Slide Time: 4:58)



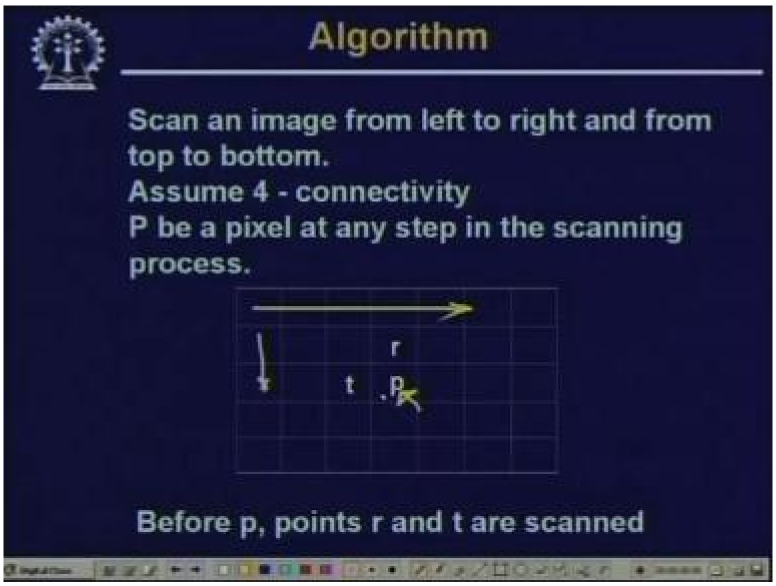
So, the problem of connected component labeling is something like this. Here, we want to assign some label to each of the pixels or the label indicates that to which particular object that particular pixel belongs. So, the object pixels belonging to a particular object should obtain the same label whereas the pixels belonging to different objects should be labeled with different numbers.

So, the connected component labeling problem is the ability to assign different labels to various disjoint connected components of an image. So, as has been shown in the particular example; here, in this particular case, we have got 2 different object regions. One object region is this and the other object region is this. So, after connected component labeling, we should be able to say that this, this, this and so the other pixels belong to one particular region. Similarly, this pixels belongs to another particular object. So, after component labeling, we should be able to patrician these pixels in 2 different course as is shown here in yellow color and red color.

So, the pixels having yellow color, they belong to one particular object and pixels having a red color, they belong to another object. So once I identify this belongingness of a pixel to a particular object, I can find out the different properties of that particular region as we have all ready said; we can find out what is the shape of the particular region, we can find out what is the area of the object, we can find out what is the boundary of this object and also we can find out different other shape area or boundary based features.

So now, let us see that what should be the algorithm for connected component labeling. To tell you the algorithm, what is done is given an image; now because we have distinguished between the pixels belonging to the object and the pixels belonging to the background, so the image now will be a binary image, there will be a set of points having value equal to 1 and a set of points having value equal to 0. So, the pixels with value equal to one belongs to an object, whereas pixels with value equal to 0 belong to the background. So, for component labeling the approach taken is like this you scan the image that is the binary image in a raster scan fashion that is from left to right and from top to bottom.

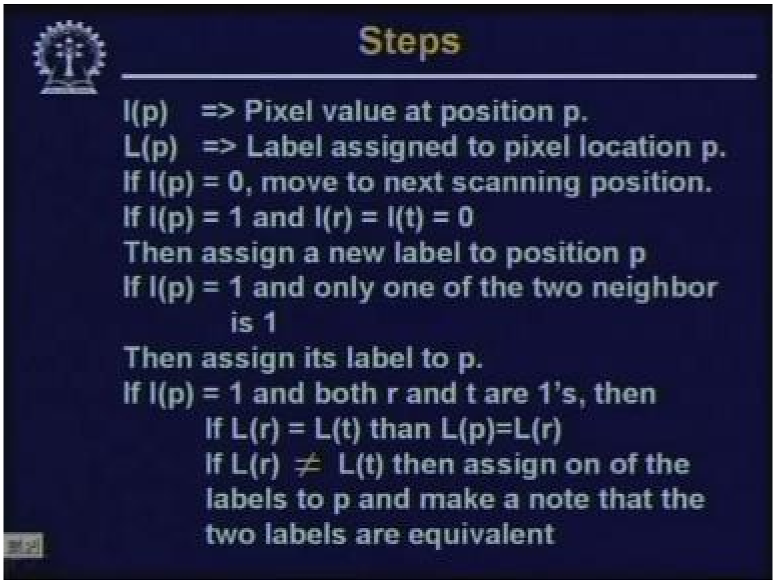
(Refer Slide Time: 7:59)



So, given in this form, you find that if we scan this image from left to right and from top to bottom, then what we get is something like this; if I scan this image from left to right and then top to bottom and while scanning, we want to level the different points in this image with a particular number which is the leveling problem. Now in this particular case, if we try to find out what should be the level at this particular point p, here I assume that p is an object point; then following the way of our scanning, you find that before p is scanned, the above points which belong to four neighbors of this point p which will be scanned or one point above p in the same column and one point to the left of p in the same row.

So, those points here are point r and point t. So, r is above p and t is to the left of p and these are the only points belonging to the 4 neighbor of p which will be scanned before the point p is scanned.

(Refer Slide Time: 9:22)



So here, our leveling algorithm will be like this that if I find if I want to label the point p, I assume that L to be I (p) is the intensity at value at the location p and L (p) is the label assigned to the pixel at location p. Now obviously, if I (p) is equal to 0 that means point p belongs to the background. So, I do not need to label that particular point p. So similarly, I move to the next point following the same scanning order. But if the intensity at point p is equal to 1, then I have to level that particular point.

So here, I check that what is the intensity values at r and t. If both at r and t, the intensity values are equal to 0; then I assign a new level to that particular point p. So, if L (p) equal to 1 and both I (p) equal to 1 and both I (r) and I (t) they are equal to 0; then I assign a new level to pixel position p. But if I (p) equal to 1, but one of its neighbors that is either r or t is having a value equal to 1 because r and t are already scanned before scanning the point p. So, here we find that either r or t whichever is having a value equal to 1 that will all ready be labeled.

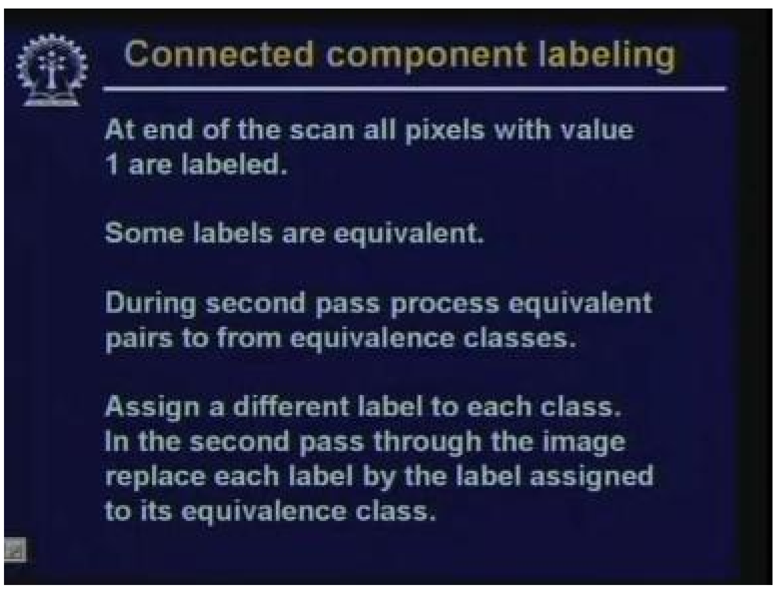
So in this case, you assign the same level to the point p. If I (p) equal to 1 and both r and t are ones, then I have 2 situations. That is the level at r and level at t, they are same. So, if they are same, you assign the same level to point p. But the problem comes if the level at r and level at t, they are different. So, both r and t are the object points but the levels are different.

So in this case, when I assign a level to the particular point p, I have to choose one of those two levels; either the level of t or the level of r which will be assigned to point p. Now, whichever level I assigned to point p, I have to make an association with other level saying that the level which is assigned to point p and the other level which is not assigned to point p, they are equivalent.

So once I do this; after this pass of the algorithm, what I have is again the binary image but now the different object points will be having different levels and at the same time, I will also have a set of equivalent pairs. So, what I have to do is I have to process all this equivalent pairs to find out the equivalent process. That is all the levels which are equivalent following the transitivity relation and then to each of the equivalence class, I have to assign a unit level.

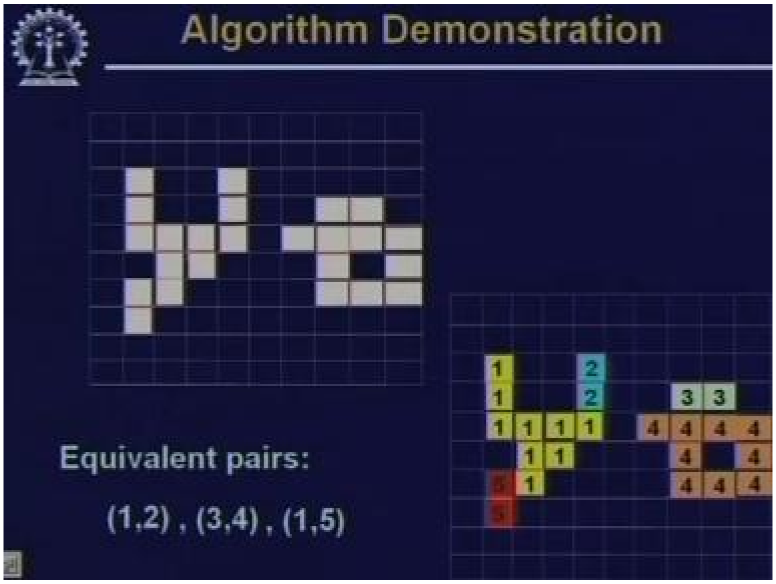
And, after that I have to do a second pass over the leveled image and all the levels which belongs to a particular equivalent class, I have to replace its original level to the level which is assigned to the equivalence class.

(Refer Slide Time: 12:47)



So, find that this component leveling algorithm in this case is a 2 pass algorithm. In the first pass, you assign some level and you generate some equivalence pairs or equivalent relations and in the second pass, you do the final leveling of the object points where different pixels belonging to a same region will get the same level.

(Refer Slide Time: 13:12)



So now, let us see how the algorithm works. You find that in this case, we have shown 2 different regions of connected pixels. So, the objects the pixels belonging to a particular object are marked with white with color white and the pixels belonging to the background, they are marked with color black.

Now, if I scan this particular binary image from left to right and from top to bottom then and I assign the levels to different pixels; then what I get is the first object point that I encounter gets a level 1. Then I continue my scanning, go to the next point belonging to the object and here you find that the next point gets a level 2 because none of the points above it or below has got any level.

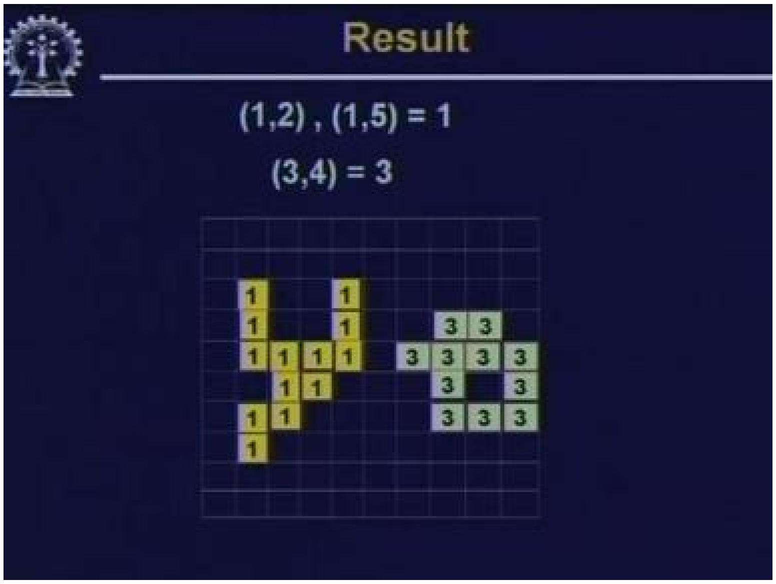
I continue further, I come to the third point and you find that for the third point, the point above it was all ready having a level 1. So, I assign the same level 1 to this particular point. Continue further, the next point gets level 2 because the point above 8 had a level equal to 2. Next point gets a new level and in this case, the level is equal to 3 because when I assign the level to this particular point, I find that none the points above it or to the left of it have got any level. So, this point gets a new level and which is equal to 3 in this particular case.

If I continue, you find that the next point again gets level 3. Continue further; the next point gets level 1, the next one also gets level 1, next one also gets level 1. Now, comes the problem; when I go to the next point, you find that the point above it had a level equal to 2 and point to the left of it had a level equal to 1. So in this case, we have given level 1 to this point. But once I have given level 1 to this point, now find that this 1 and 2, they are equivalent. So, after leveling this point with level equal to 1, I have to set that 1 and 2 are equivalent and that has to be noted. So, I note that 1 and 2, they are equivalent.

You continue further; next point gets level equal to 4, next point also gets level equal to 4. Again, I get the same situation that the point above it had a level equal to 3. So, I have to note that 3 and 4 are equivalent. So, the equivalence between 3 and 4 are noted. I continue this way; next point gets 4, next point also gets 4, next point gets level 1, next point level 1, next point gets level 4, the next point also gets 4, then I get a level 5, the next point gets a level 1 but at this point again I have to note that 5 and 1 are equivalent.

So, if I continue this, I get the level of this point. So, I get at the end of the first scan, 3 equivalent pairs; (1, 2) (3, 4) and (1, 5). So, next what I have to do is I have to process this equivalent pairs to find out what are the equivalence classes.

(Refer Slide Time: 16:57)



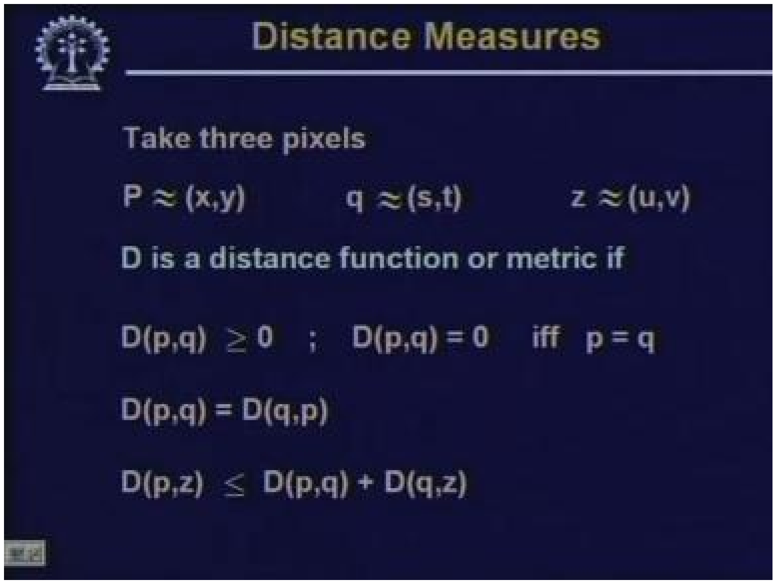
So, that is what I do next; so if I process the equivalent pairs 1 and 2 and 1 and 5; I find that all 1, 2 and 5, they are equivalent and here I assign a level 1 to this equivalence class containing levels 1, 2 and 5. Similarly 3 and 4, they are also equivalent and here I assign the level 2 level 3 to this particular class (3, 4).

So, after this new levels are assigned to the equivalence classes, I do a second pass over the same leveled image and every level is now replaced by the label assigned to its equivalence class and at the end of the second pass, you find that what I get is leveled regions in the image. Now, the 2 regions are clearly identifiable. One region get level equal to 1 and the other region gets level equal to 3.

So once I get this, I know that which pixel or which point belong to which particular object. All the points having level equal to 1, belong to one object and all the points having level equal to 3 that belong to some other object. So, now by identifying the levels; I can find out what is the shape of different objects, what is the area of different objects and many search shape related features that we can extract after this leveling.

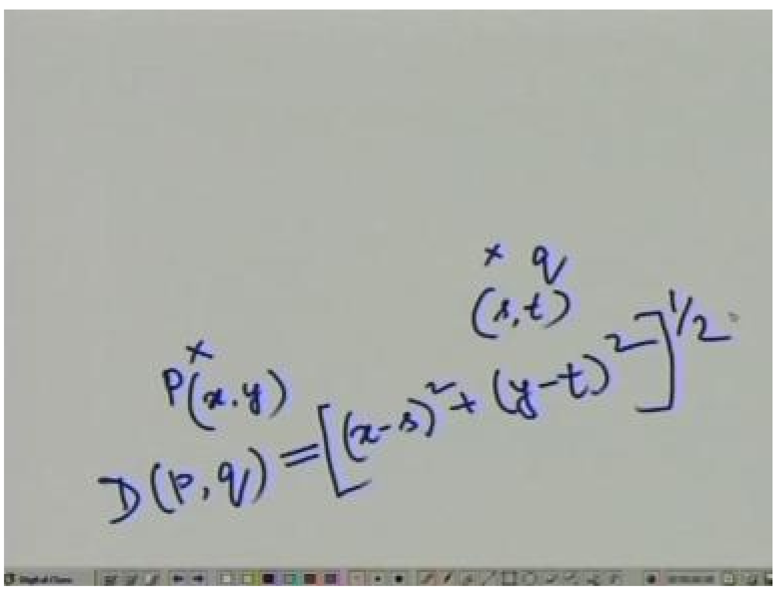
So, find that this component leveling algorithm or component leveling operation is a very very important operation which is useful for a high level image understanding purpose.

(Refer Slide Time: 18:53)



So, after doing this component leveling algorithm, now let us move to another operation, another concept that is distance measures. Now, finding out the distance between 2 points, we are all familiar that if I know the coordinate or the location of 2 different points, I can find out what is the distance between the 2 points.

(Refer Slide Time: 19:16)



Say for example, what I can do is if I have 2 points, say one point P and other point q and I know that the coordinate of point P is given by (x, y) and the coordinate of point q is given by s and t; then we all know from our school level mathematics that the distance between the 2 points P and q is given by the relation that I represent this as D (p, q) that is distance between p and q which

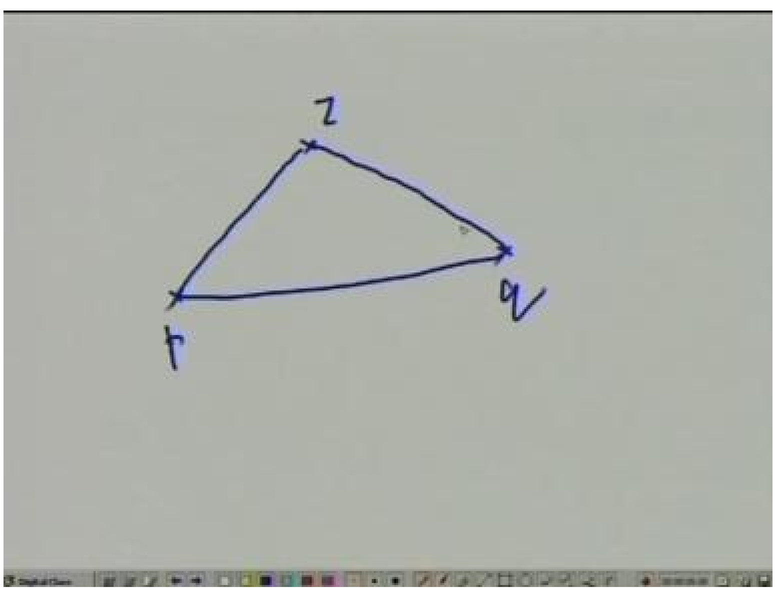
will be given by x minus s square plus y minus t square and square root of this term. So, this is what all of us know from our school level mathematics.
Now, when I come to digital domain, then this is not only this is not the only distance measure that can be used. There are various other distance measures which can be used in digital domain. Those distance measures are say; city block distance, chess board distance and so on. So, to see that if D is a distance function or a distance metric, then what is the property that should be followed by this distance function D?

So for this, let us take 3 points. We take 3 points here; p having a coordinate (x, y), q having a coordinate (s, t) and I take another point z having the coordinate (u, v). Then D is called a distance measure is a valid distance measure or valid distance metric. If D (p, q) is greater than or equal to 0 for any p and q, any 2 points p and q; D (p, q) must be greater than or equal to 0 and D (p, q) will be 0 only if p is equal to q.

So, that is quite obvious because the distance of the point from the point itself has to be equal to 0. Then the distance metric distance function should be symmetric that is if I measure the distance from p to q, that should be same as the distance if I measure from q to p. That is the second property that must hold true. That is D (p, q) should be equal to D (q, p).

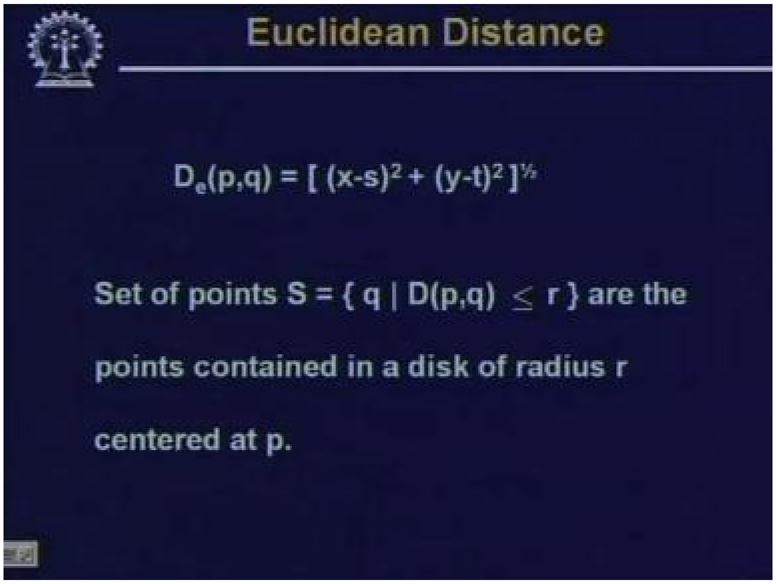
And, there is a third property which is an inequality. That is if I take a third point z, then the distance between p and z that is D (p, z) must be less than or equal to the distance between the p and q plus the distance between q and z and this is quite obvious, again from our school level mathematics you know that if I have say 3 points (p, q) and I have another point z and if I measure the distance between p and z, this must be less than the distance between pq plus the distance between pz.

(Refer Slide Time: 22:30)



So, this what we all have done in our school level mathematics and the same property must hold true if an in case of digital domain where we talk about different other distance functions. So, these are the 3 properties which must hold true for a function if the function is to be considered as a distance function or a distance metric.

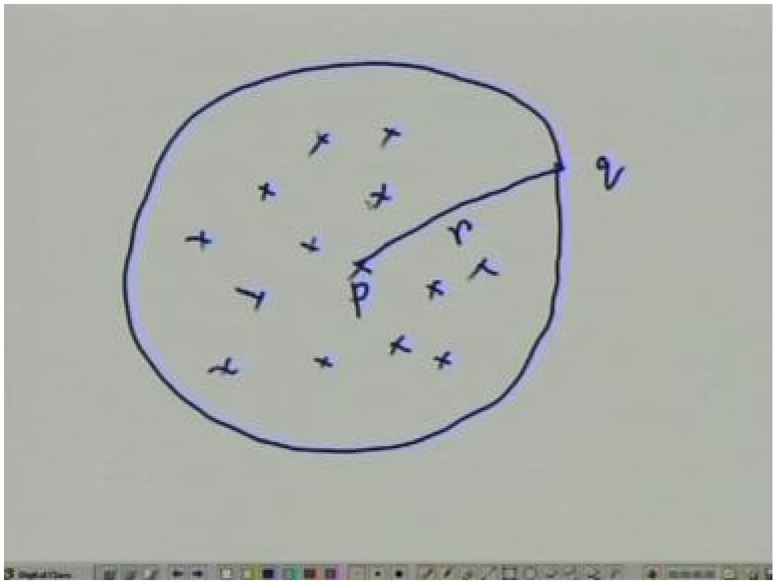
(Refer Slide Time: 23:23)



Now, the first of this, that is the distance between p and q which we have all ready seen that if p has a coordinate (x, y) and q has a coordinate (s, t), then D (p, q) the distance between p and q is equal to x minus s square plus y minus t square and square root of this whole term. This is a distance measure which is called a Euclidean distance.

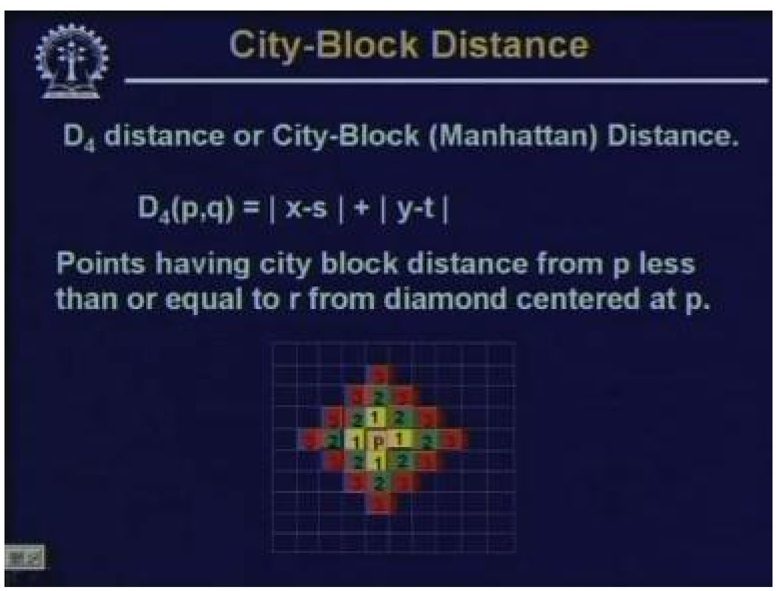
So, in case of Euclidean distance, you find that set of points q where D (p, q) the distance between p and q, obviously we are talking about the Euclidean distance is less than or equal to some value r. So, set of all this points are the points contained within a disk of radius r where the center of the disk is located at location p.

(Refer Slide Time: 24:19)



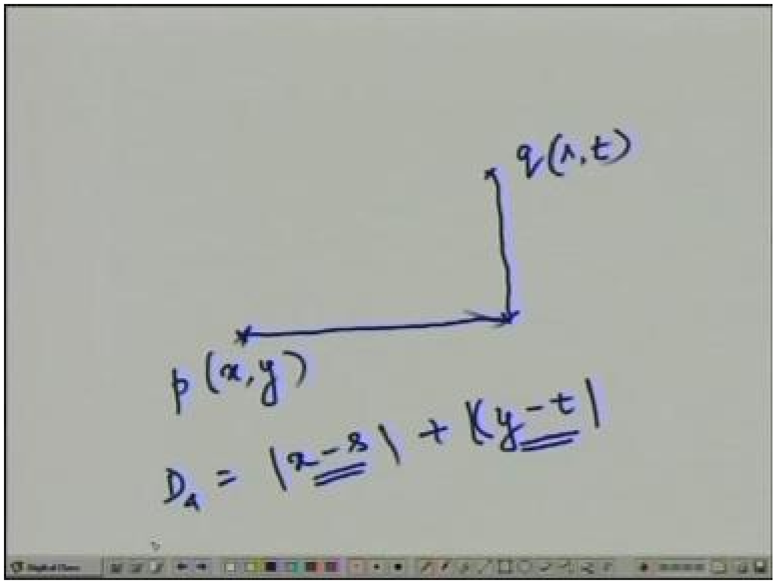
And again, this is quite obvious, you find that suppose I have a point p here and I take a point q and I say that the distance between p and q is r. So, if I take set of all these points are the distance is equal to r that forms a circle like this. So, all other points having a distance less than r from the point p will be the points within this circle. So, set of all this points or the distance value is less than or equal to r obviously we are taking about the Euclidean distance; in that case, the set of all this points forms a disk of radius r or the center of the disk is at location p.

(Refer Slide Time: 25:11)



Now, coming to the second distance measure which is also called D4 distance or city block distance or this is also known as Manhattan distance; so this is defined as D_4 (p, q) is equal to x minus s absolute value plus y minus t absolute value.

(Refer Slide Time: 25:44)



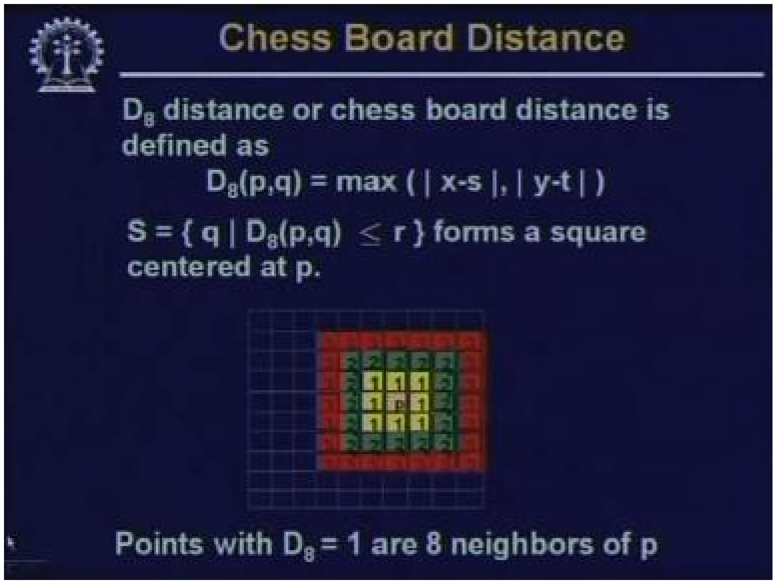
So, in this case you find that it is some like this that if I have point p with coordinate (x, y) and I have point q with coordinate (s, t). So, this D_4 distance as it is defined that it is equal to x minus s, take the absolute value plus y minus t, again you take the absolute value.

So, this clearly indicates that if I want to move from point p to point q, then how much distance I have to move along the x direction and how much distance I have to move along the y direction. Because x minus s, the absolute value of this is the distance travelled along x direction and y minus t absolute value of this is the travel is the distance travelled along the y direction.

So, the sum of these distances along x direction and y direction gives you the city block distance that is D4. And, here you find that the points having a city block distance from point p less than or equal to some value r, will from a diamond center centered at point p. So, which is quite obvious from here that here you find that if p is the point at the center, then all the points having city block distance, they are just the 4 neighbors of the point p.

Similarly, all the points having the city block distance is equal to 2, they are simply the points which are at distance 2. That is the distance taken in the horizontal direction plus the distance taken in the vertical direction that becomes equal to 2 and set of all this points with city block distance is equal to 2 that simply forms a diamond of radius 2 and similarly other points at distances 3 4 and so on.

(Refer Slide Time: 28:13)



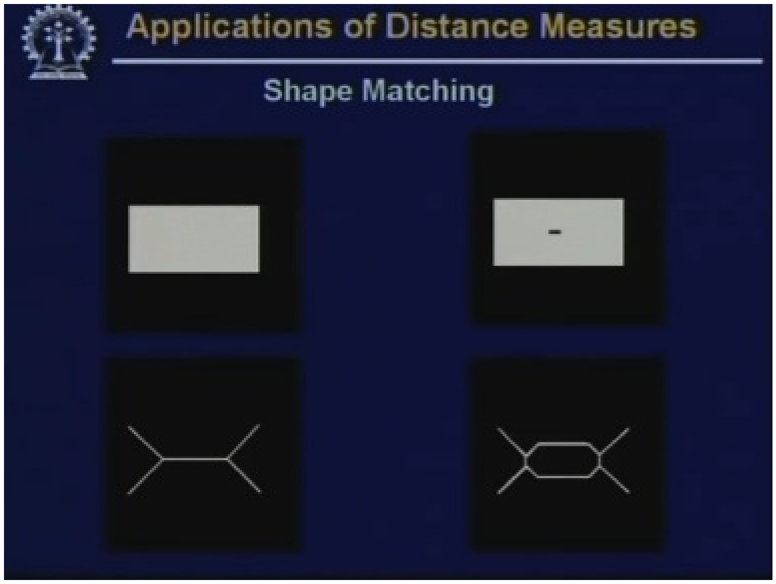
Now, we come to the third distance measure which is the chess board distance. As you have seen that in case of city block distance, the distance between 2 points was defined as the sum of the distances that you cover along x direction plus the distance along the y direction. In case of chess board distance, it is the maximum of the distances that you cover along x direction and y direction.

So, this is D_8 (p, q) which is equal to max of x minus s and y minus t where we take the absolute value of both x minus s and y minus t and following the same argument, here you find that the set of points with a chess board distance of less than or equal to r, now forms a square centered at point p. So here, all the points with a chess board distance of equal to 1 from point p, they are nothing but the 8 neighbors of point p.

Similarly, the set of points with a chess board distance will be equal to 2 will be just the points outside the points having a chess board distance equal to 1. So, if you continue like this you will find that all the points having a chess board distance of less than or equal to r from a point p will form a square with point p at the center of the square. So, these are the distance different distance measures that can be used in the digital domain.

Now, let us see that what is the application of this distance measure. One of the obvious applications is that if I want to find out the distance between 2 points, I can make use of either Euclidean distance or city block distance or the chess board distance. Now, let us see one particular application other than just finding out the distance between 2 points.

(Refer Slide Time: 30:14)



Say for example, here I want to match 2 different shapes which are shown in this particular diagram. Now, here you find that these 2 shapes are almost similar except that you have a hole in the second shape. But if I simply go for matching these 2 shapes, they will be almost similar. So, just by using these original figures, I cannot possibly distinguish between these 2 shapes.

So, if I want to say that these 2 shapes are not same that they are dissimilar; in that case, I cannot work on this original shapes but I can make use of some other feature of this particular shape. So, let us see what is that other feature. If I take the skeleton of this particular shape; in that case, you find that this third figure gives you what is the skeleton of the first shape. Similarly, the fourth figure gives you what is the skeleton of the second shape.

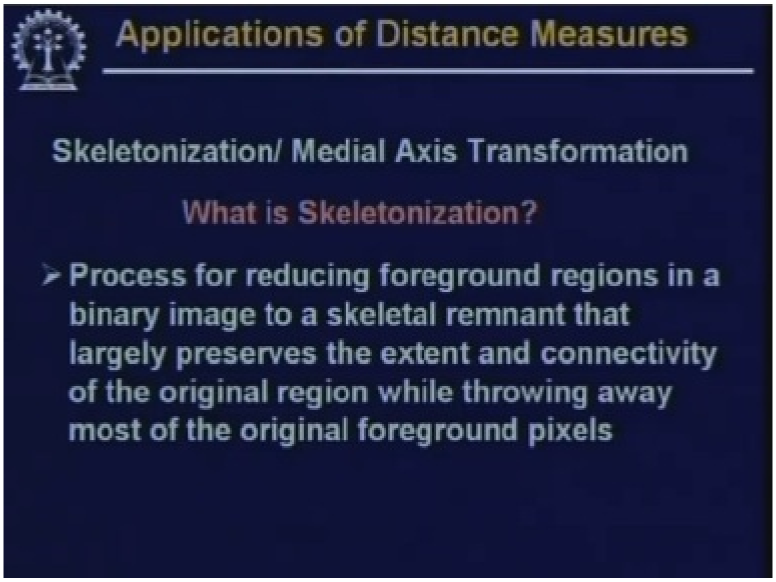
Now, if I compare these 2 skeletons rather than comparing the original shapes, you will find that there is lot of difference between these 2 skeletons. So, I can now describe the shapes with the help of the skeletons in the sense that I can find out that how many line segments are there in the skeleton. Similarly, I can find out that how many points are there where more than 2 line segments meet.

So by this, if I compare the 2 skeletons, you find that for the skeleton of the first shape, there are only 5 line segments, whereas for the skeleton of the second shape there are 10 line segments. Similarly, the number of points where more than 2 line segments meet; in the first skeleton there are only 2 such points, whereas in the second skeleton there are 4 such points.

So, if I compare using the skeleton rather than comparing using the original shape, you find that there is lot of difference that can be found out both in terms of the number of line segments, the skeleton has and also in terms of the number of points where more than 1 line segments meet.

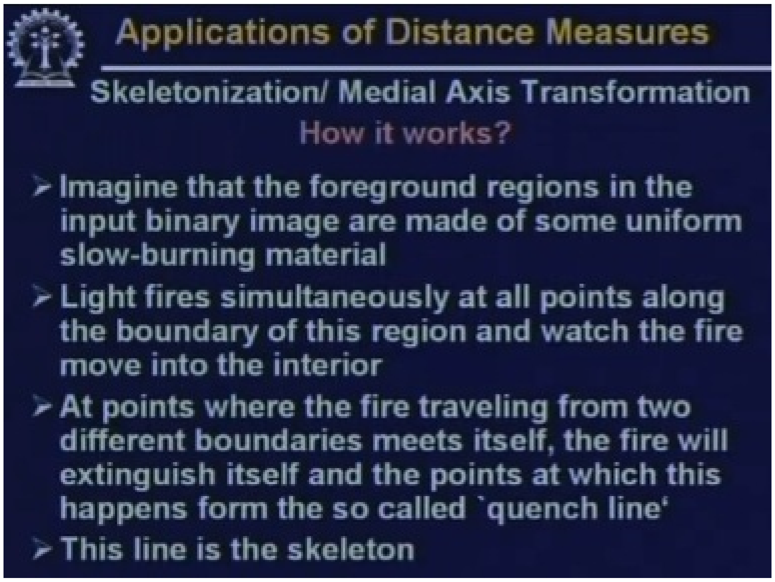
So using these descriptions which I have obtained from the skeleton, I can distinguish between the two shapes as shown in this particular figure. Now, the question is that how do we get this skeleton and what is this skeleton.

(Refer Slide Time: 33:01)



So, you find that if you analyze the skeletons, you will find that the skeletons are obtained by removing some of the foreground points but the points are removed in such a way that the shape information as well as the dimension likely that is what is the length of rate of that particular shape is more or less retained in the skeleton. So, this is how this is what is the skeleton of the particular shape and now the question is how to obtain the skeleton.

(Refer Slide Time: 33:41)

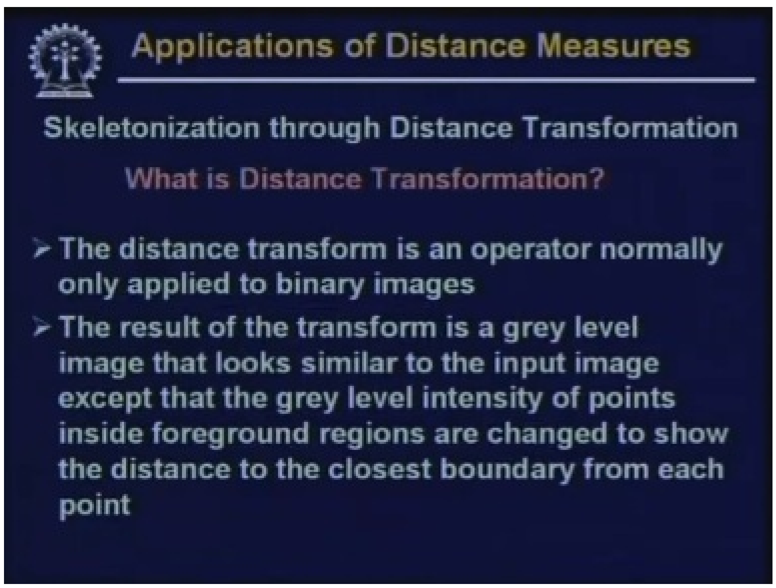


Now, before coming to how do you obtain the skeleton; let us see that how the skeleton can be found out. So, the skeleton can be found out in this manner; if I assume that the foreground region in the input binary image is made of some uniform slow burning material and then what I do is I light fire at all the points across the boundary of this region that is the foreground region.

Now, if I light fire across the boundary points simultaneously, then the fire lines will go in slowly because the foreground region consists of slow burning material, then you will find that as the fire lines they go in, there will be some points in the foreground region where the fire coming from 2 different boundaries will meet and at that point, the fire will extinguish itself.

So, the set of all those points is what is called the quench line and the skeleton of the region is nothing but the quench line that we obtained by using this fire propagation concept.

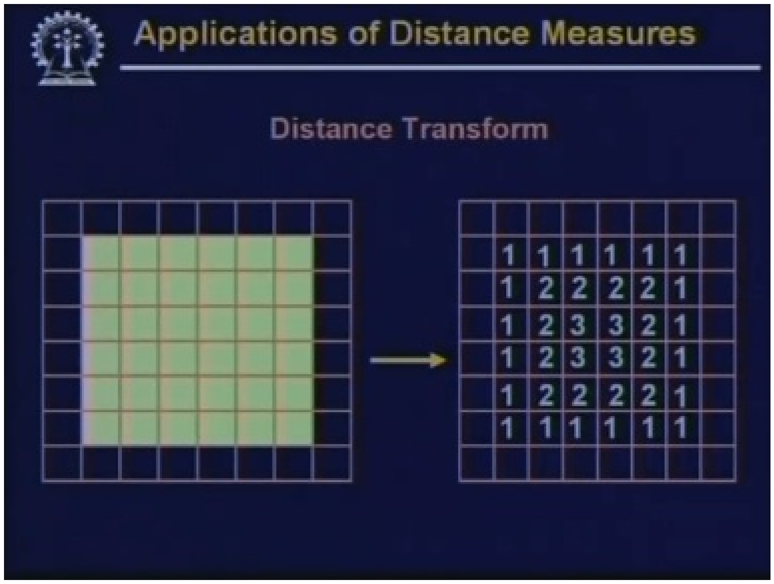
(Refer Slide Time: 35:04)



Now, to obtain this kind of skeleton, you find that this simple description of movement of the fire line does not give you an idea of how to compute the skeleton of a particular shape. So, for that what we can use is something called distance measure. Now, the distance measure is in the same manner. We can define that when we a lighting the fire across all the boundary points simultaneously and the fire is moving inside the foreground region slowly, we can note at every point that how much time the fire takes to reach that particular point that is the minimum time the fire takes to reach that particular point and at every such foreground point if we note this time taken to reach, the time the fire takes to reach that particular point; then effectively what we get is a distance transformation of the image.

So, in this case, you find that distance transform is normally used for binary images and because at every point we are noting the time the fire takes to reach that particular point; so by applying distance transformation, what we get is an image or the shape of the image is similar to the input binary image but in this case, the image itself will not a binary but it will be a grey level image where the grey level intensity of the points in the inside the foreground region had changed to show the distance of that point from the closest boundary point.

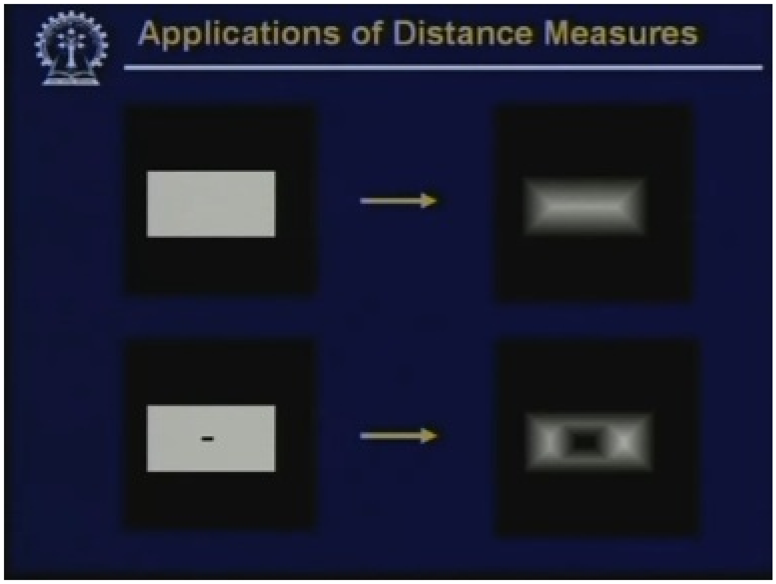
(Refer Slide Time: 36:46)



So, let us see that what this distance transform means. Here, you find that here we have shown a particular binary image where the foreground region is a rectangular region and if I take the distance transform of this, the distance transformed image is shown in the right hand side. Here you find that all the boundary points, they are getting a distance value equal to 1. Then the points inside the boundary points, they get a distance value equal to 2 and the points further inside, they gets a distance value equal to 3.

So, you find that the intensity value that we are assigning to different points within the foreground region, the intensity value increases slowly from the boundary to the interior points.
So, this is nothing but a grey level image which you get after performing the distance transformation.

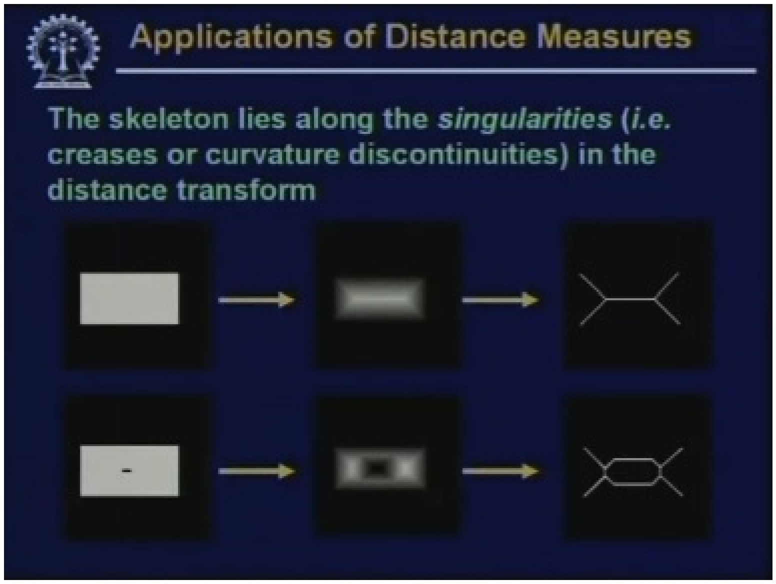
(Refer Slide Time: 37:52)



So now, you find that if I apply the distance transformation to the shapes that we have just discussed, the 2 rectangular shapes; on the left hand side, we have the original rectangular shape or the binary image and on the right hand side, what is shown is the distance transformed image and here you find that again in this distance transformed image as you move inside, inside the foreground region, the distance value increases gradually.

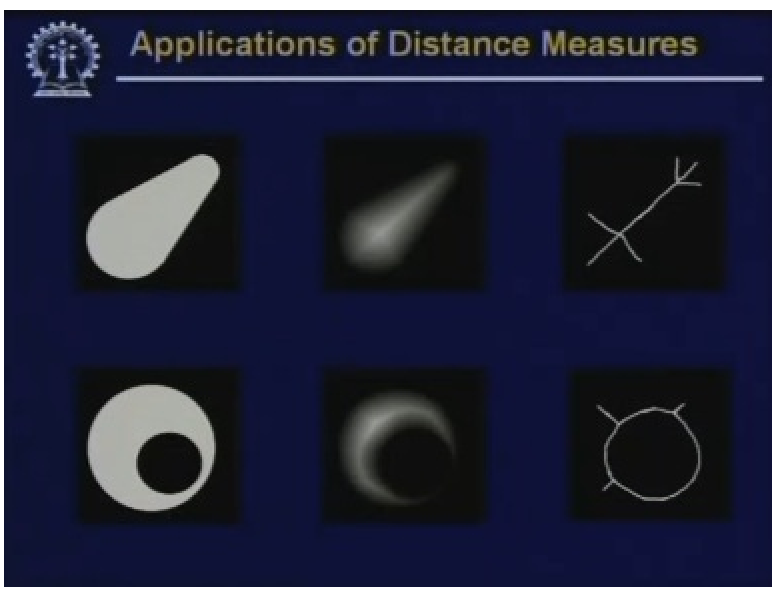
And now, if you analyze this distance transformed image, you find that there are few points at which there is some discontinuity of the curvature. So, from this distance transformed image, if I can identify the points of discontinuity or curvature discontinuity, those are actually the point which lies on the skeleton of this particular shape.

(Refer Slide Time: 38:46)



So, as shown in the next slide, you find that on the left hand side, we have the original image; the middle column tells you the distance transformed image and the right most columns tells you the skeleton of this particular image. And, if you now correlate this right most column with the middle column, you find that the skeleton in the right most columns can now be easily obtained from the middle column which tells you that what is the distance transform of the shape that we have considered.

(Refer Slide Time: 39:24)

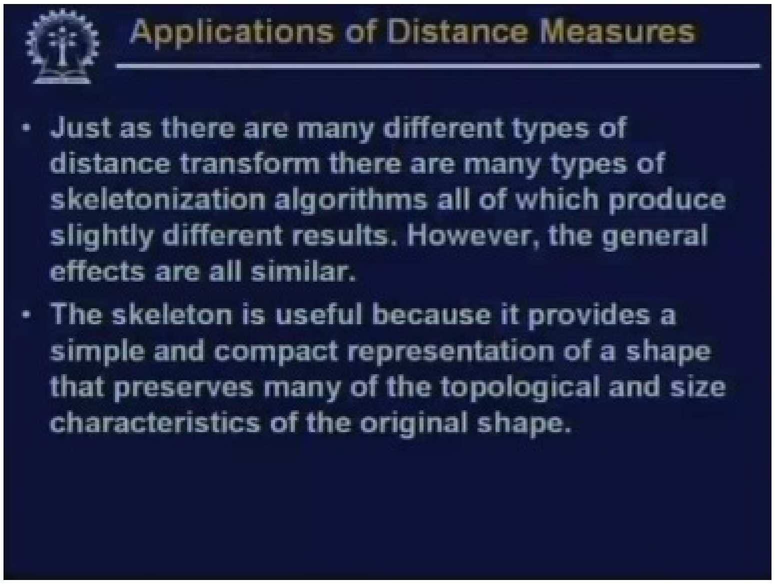


So, this shows some more skeletons of some more shapes. Again, on the left hand side we have the original image, in the middle column we have the distance transformed image and on the right most columns we have the skeleton of this particular shape. So, here again you can find that the relation between the skeleton and the distance transformed image, they are quit prominent. So, for all such shapes or whenever we go for some shape matching problem or shape discrimination problem; in that case, instead of processing on the original shapes if we compare the shapes using this skeleton, then the discrimination will be better than if we compare the original shapes.

Now here, when I am going for a distance transformation of a particular shape, as we have seen that we can have different types of distance measures or distance metrics like Euclidian distance metric, we can have city block distance metric or if an we can have a chess board distance metric; similarly, when I take this distance transformation for each of the distance metrics, there will be different transformations, different distance transformations.

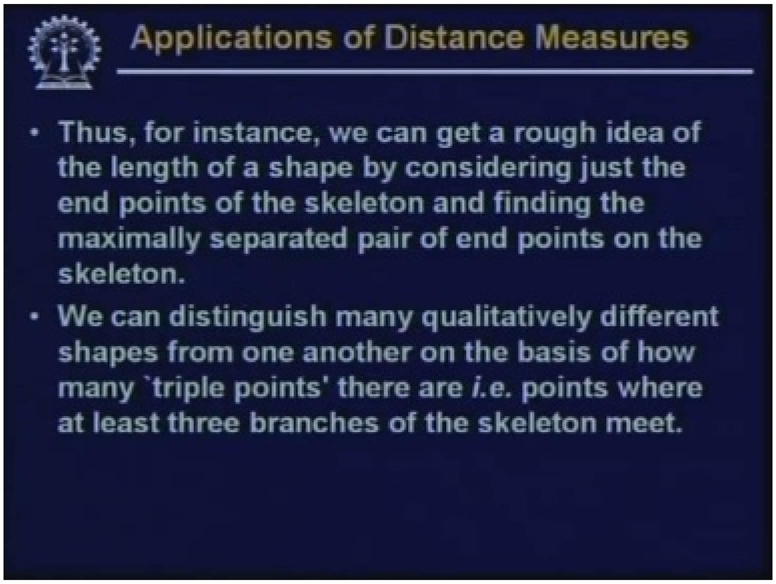
And obviously, the difference different distance transformations will produce different results but all the results that we will get similarly from the distance transformed image when you get the skeleton, all the skeletons that we will get using different distance metrics, they will be a slightly different but they will be almost similar. So, this can be just another application of the distance metric.

(Refer Slide Time: 41:13)



And, you find that here the skeleton is very very useful because it provides a simple and compact representational shape that preserves many of the topological and size characteristics of the original shape.

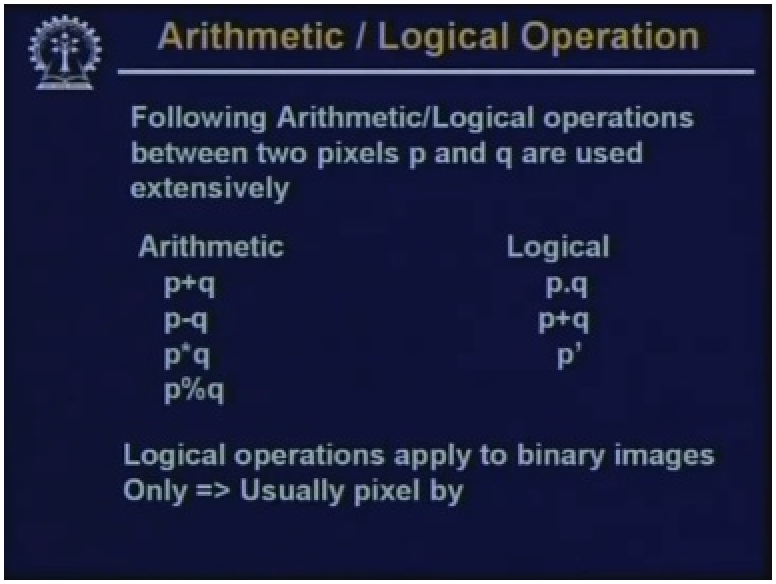
(Refer Slide Time: 41:34)



Also, from this skeleton, we can get a rough idea of the length of a shape because when I get the skeleton, I get different end points of the skeleton and if I find out the distances between every pair of end points in the skeleton, then the maximum of all those pair wise distances will give me an idea of what is the length of that particular shape and as we have all ready said that using this distance using this skeleton, we can qualitatively differentiate between different shapes because here we can find out we can get a description of the shape from the skeleton in terms of the number of line segments that the skeleton has and also in terms of the number of points in the skeleton were more than 2 line segment meet.

So, as we have said that the distance metric of the distance function is not only useful for finding out the distance between 2 points in an image but the distance metric is also used for useful for other applications. Though here also, we have found out distance measure between different pair of points and for skeletonization what we have used is we have first taken the distance transformation and in case of distance transformation, we have taken the distance of every foreground pixel from its nearest boundary pixel and that is what gives you a distance transformed image and from the distance transformed image, we can find out the skeleton of that particular shape considering the points of curvature discontinuity in the distance transformed image. And, latter on also we will see that this distance metric is useful in many other cases.

(Refer Slide Time: 43:33)

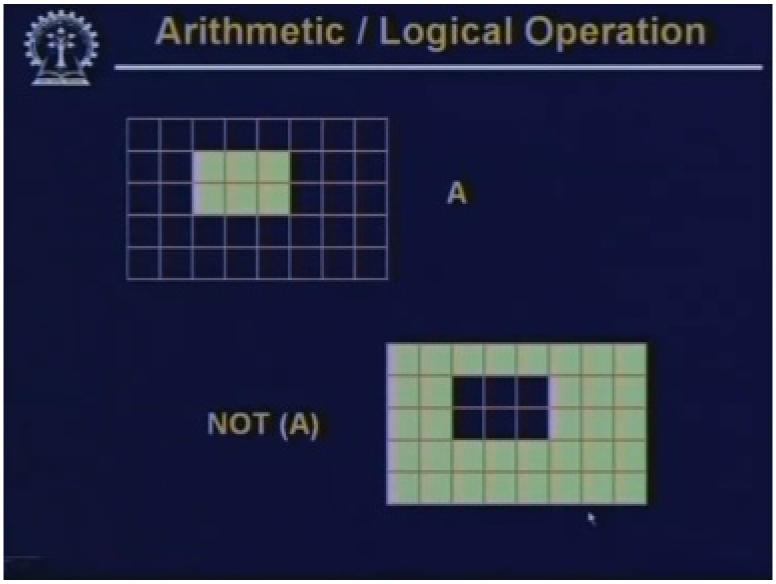


Now, after our discussion on this distance metrics, let us see that what are simple operations that we can perform on the images. So, as you have seen that in case of numerical system, whether it is decimal number system or binary number system; we can have arithmetic as well as logical operations. Similarly, for images also we can have arithmetic and logical operations.

Now coming to images, I can add 2 image pixel by pixel. That is a pixel from an image can be added to the corresponding pixel of a second image. I can subtract 2 images pixel by pixel that is pixel of an image can be subtracted from the corresponding pixel of that of another image. I can go for pixel by pixel multiplication; I can also go for pixel by pixel division. So, these are the different arithmetic operation that I can perform on 2 images and these operations are applicable both in case of grey level image as well as in case of binary image.

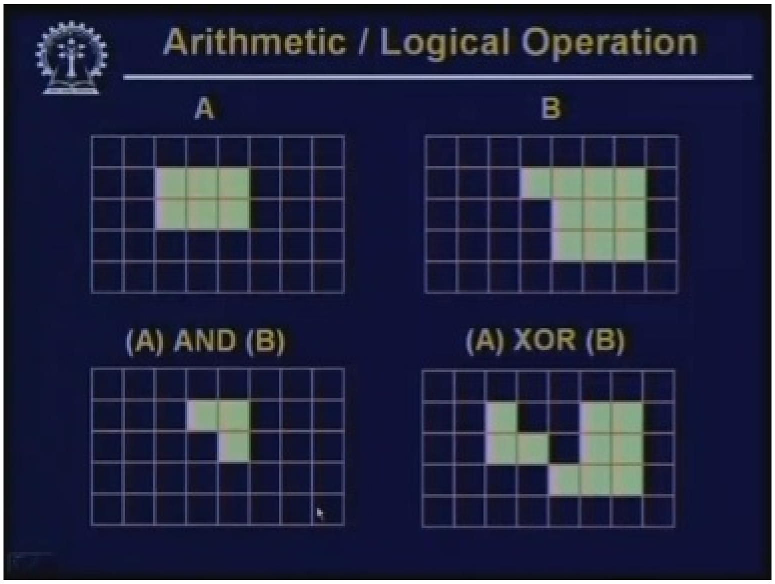
Similarly, in case of binary images, we can have logical operation; the logical operation in terms of ANDing pixel by pixel, ORing by pixel and similarly inverting pixel by pixel. So, these are the different arithmetic logical operations that we can do on grey level image and similarly logical operations on binary image.

(Refer Slide Time: 45:06)



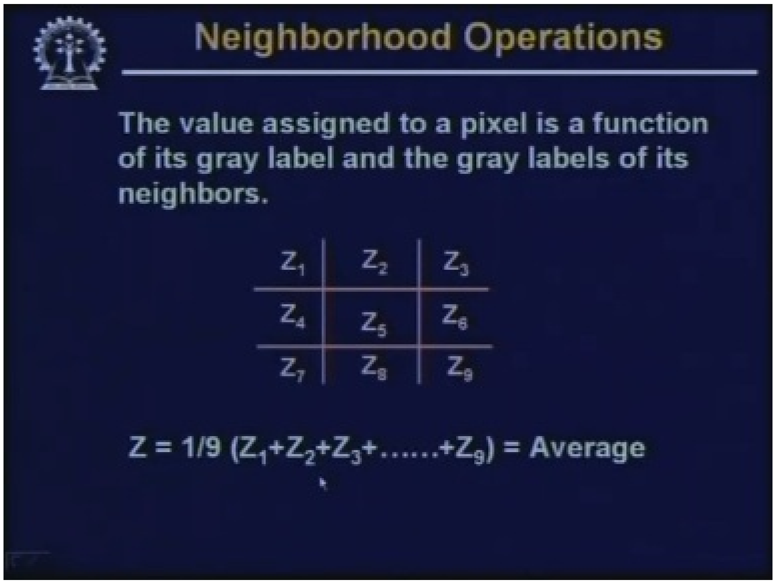
So here, is an example that if I have a binary image A where again all the pixels with value equal to 1 are shown as shown in green color and the pixels with value equal to 0 are shown in black color. Then I can just invert this particular binary image that is I can make a NOT operation or invert operation, so NOT of A is another binary image where all the pixels in the original image which was black now becomes white or 1 and pixels which are white or 1 in the original pixel, those pixels became become equal to 0.

(Refer Slide Time: 45:49)



Similarly, I can perform other operations like given 2 images - A and B, I can find out A and B, the logical ANDing operation which is shown in the left image. Similarly, I can find out the XOR operation and after XOR, the image that I get is shown in the right image.

(Refer Slide Time: 46:09)

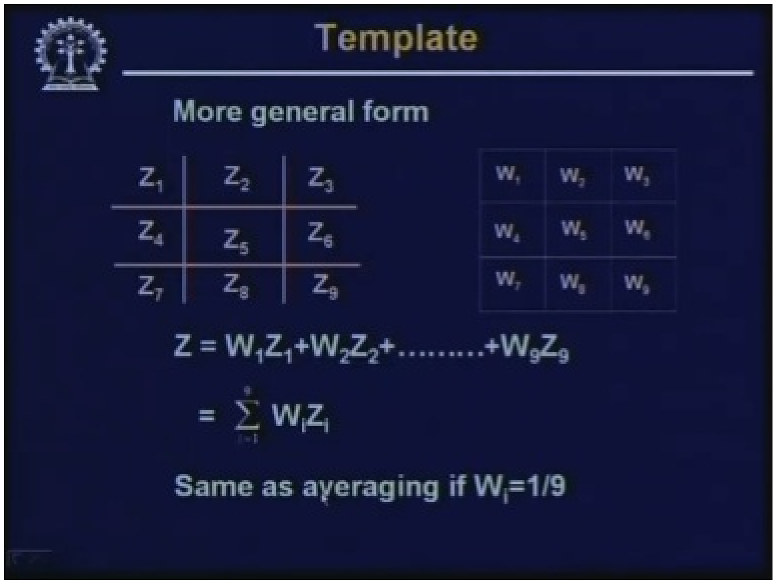


So, these are the different pixel operations or pixel by pixel operations that I can perform. In some other applications, we can also perform some neighborhood operations. That is the intensity value at a particular pixel may be replaced by a function of the intensity values of the pixels which are neighbors of that particular pixel p.

Say for example; in this particular case, if this 3 by 3 matrix, this represents the part of an image which has got nine pixel elements Z_1 to Z_9 and I want to replace every pixel value by the average of its neighborhood considering the pixel itself.

So, you find that at location Z_5 if I want to take the average, the average is simply given by Z_1 plus Z_2 plus Z_3 plus Z_4 upto plus Z_9 that divided by 9. So, this is a simple average operation at individual pixels that I can perform which is nothing but a neighborhood operation because at every pixel level, we are replacing the intensity by a function of the intensities of its neighborhood pixels. And, this averaging operation, we will see later that this is the simplest form of low pass filtering to remove noise from a noisy image.

(Refer Slide Time: 47:47)

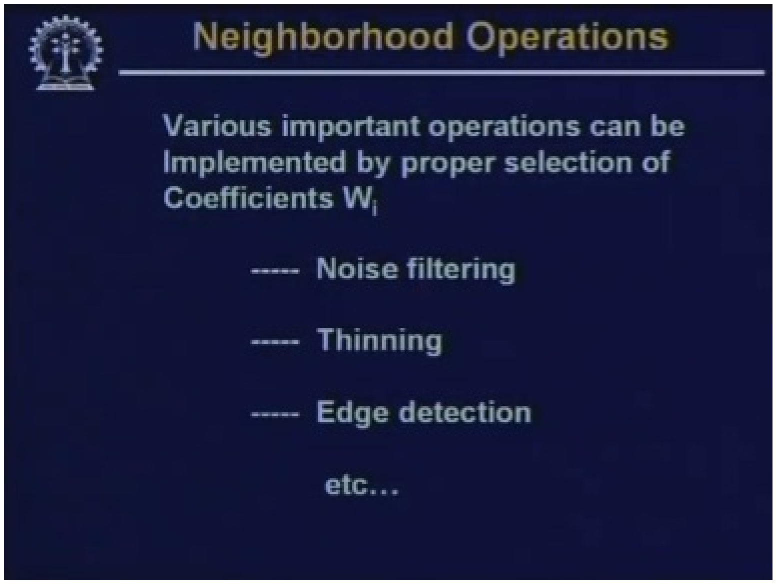


Now, this kind of neighborhood operation can be generalized with the help of templates. So here, what we do is we define a 3 by 3 template which is shown on this right hand figure where the template contains nine elements W_1 to W_9 and if I want to perform the neighborhood operation; what you do is you put this template, this particular template on the original image in such a way that the pixel at which I want to replace the value, the center of the template just coincides with that pixel.

And then at the particular location, we replace the value with the weighted sum of the values taken from the image and the corresponding point from the template. So, in this case, the value which will be replaced is given by Z equal to W_i Z_i summation of this I varying from 1 to 9. And here, you find that if I simply put W_i equal to 1 by 9 that is all the points in the template have the same value which is equal to 1 by 9, then the resultant image that I will get is nothing but the averaged image which we have done just in the previous slide.

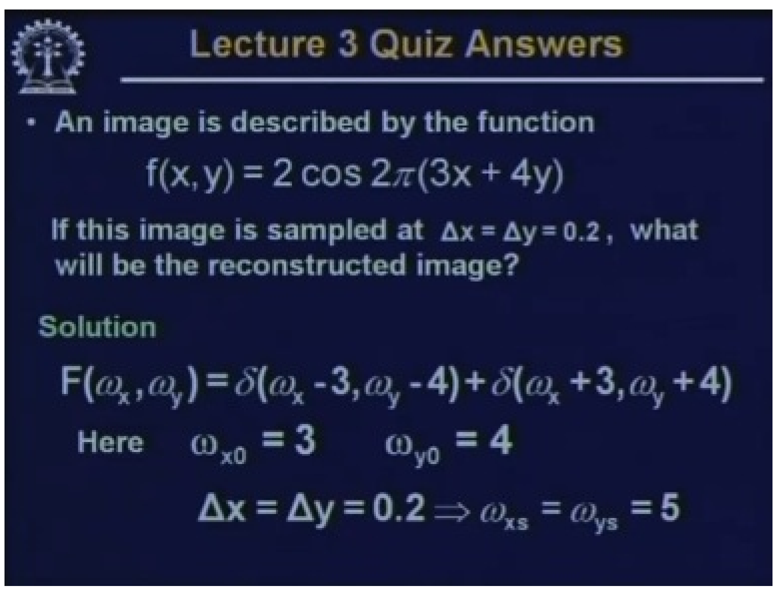
So, this neighborhood operation using the template is a very very general operation, it is useful not only for averaging purpose; it is useful for many other neighborhood operations and we will see later that this can be used for noise filtering, it can be used for thinning a binary images.

(Refer Slide Time: 49:24)



This same template operation can also be used for edge detection operation in different images. So, with this, we complete our lecture today. Now, let us see some of the solutions of the problems that we had given in lecture 3.

(Refer Slide Time: 49:52)

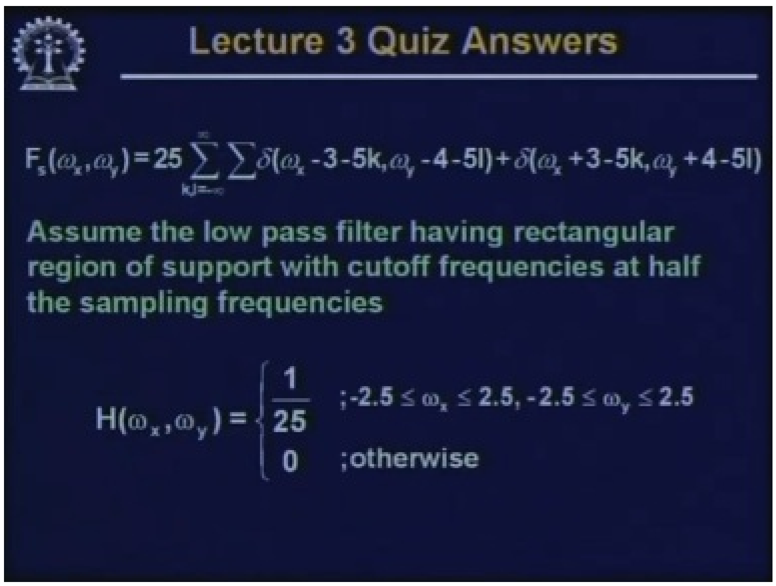


So, at the end of lecture 3, I had given a problem that an image is described by a function f (X, Y) equal to 2 cos 2 phi 3 X plus 4Y and this image is sampled at delta x equal to delta y equal to 0.2; then what will be the reconstructed image?

The solution is like this if I take the Fourier transform of the image which is given, then you will find that the Fourier transform will be a set of will be a 2 dimensional delta function which is given by del w_x minus 3 w_y minus 4 plus del w_x plus 3 w_y plus 4 and in this particular case, the maximum frequency in the X direction that is w_{x0} equal to 3 and the maximum frequency in the Y direction that is w_{y0} is equal to 4.

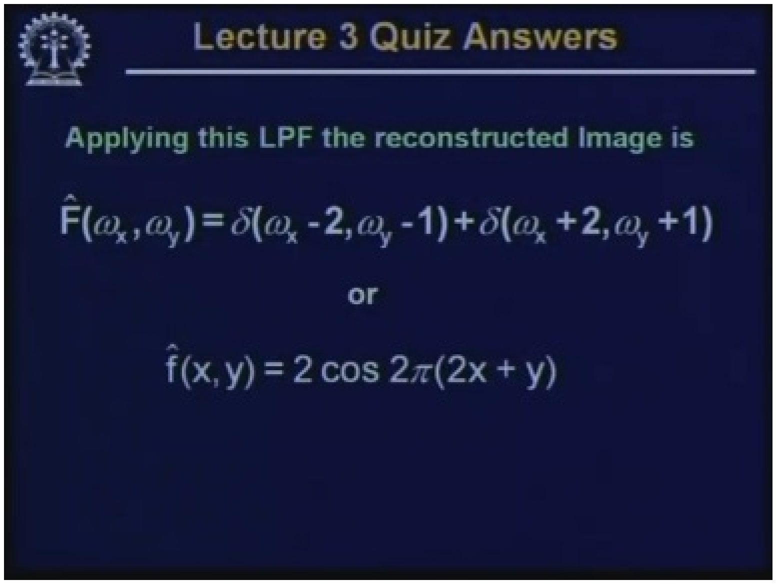
So, these are the bandwidths of the image in X direction and Y direction and here the sampling interval has been given as delta x is equal to delta y equal to 0.2. So, from this sampling interval if I calculate the sampling frequencies, you find that the sampling frequency in X direction omega $_{XS}$ will be same as omega $_{YS}$ sampling frequency in Y direction which will be equal to 5.

(Refer Slide Time: 51:23)



Now, once I have these sampling frequencies; then the sampled image, frequency spectrum of the sampled image will be as given here and given a sampling frequency for reconstruction the low pass filter that is used, usually has a bandwidth which is half of the sampling frequency. That means the low pass filter that will be used in this particular case will have a bandwidth of minus 2.5 to 2.5 both for in the Y direction, both for Y X and omega x and omega $_{Y}$.

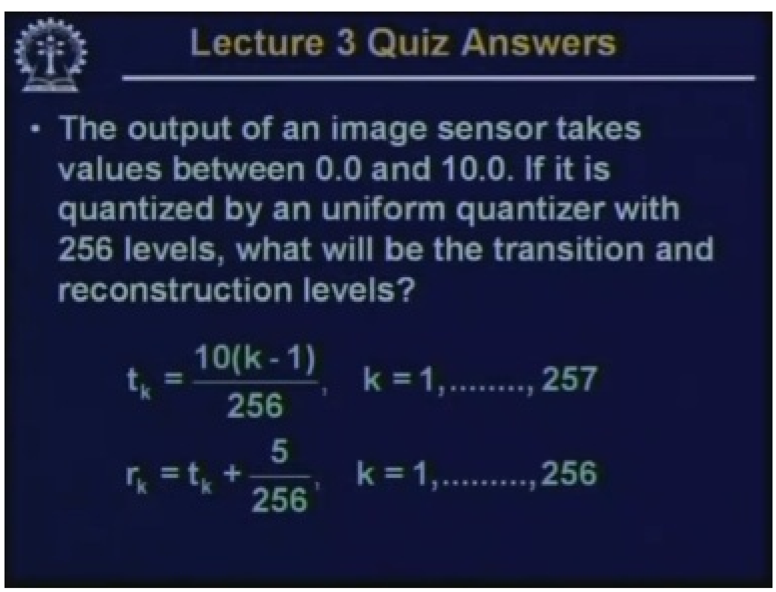
(Refer Slide Time: 52:09)



So, by using this low pass filter if you take out a particular frequency band from the sampled image, you will find that the frequency band that you will take out is nothing but at frequency 2 along X direction and frequency 1 along Y direction.

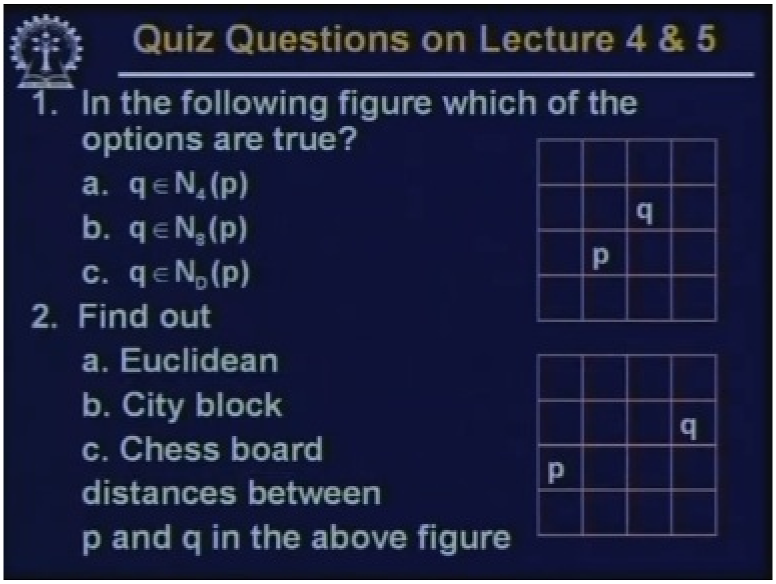
So, using this frequency band your low pass filter I mean the reconstructed image will be given by 2 cos 2 phi into 2X plus Y. So here, again, you find that naturally your reconstructed image is not same as the original image. So, lot information has been lost and the reason being that the sample frequency that we have chosen does not meet our criteria. Then we had given a second problem which is for a uniform quantizer design.

(Refer Slide Time: 52:53)



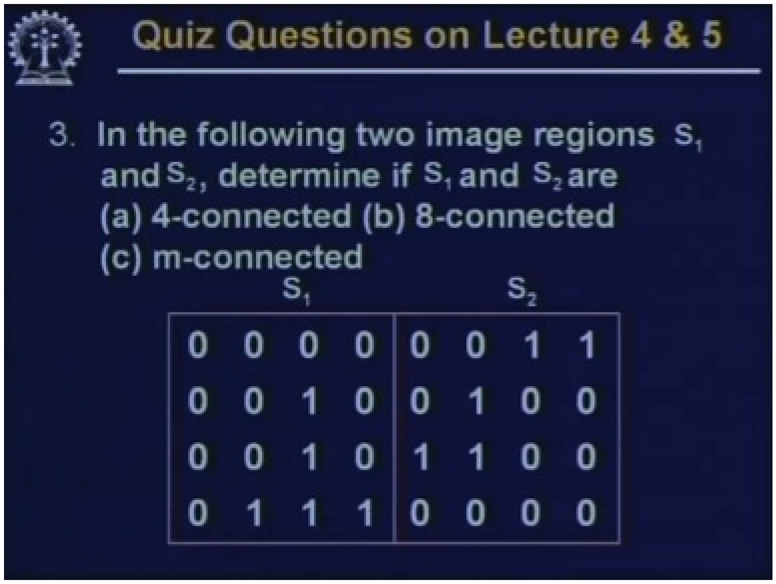
And here again, the solution is very simple. Simply follow the steps of the uniform quantizer design and you will find that the transition levels and the reconstruction levels will get as given in this 2 expressions.

(Refer Slide Time: 53:11)



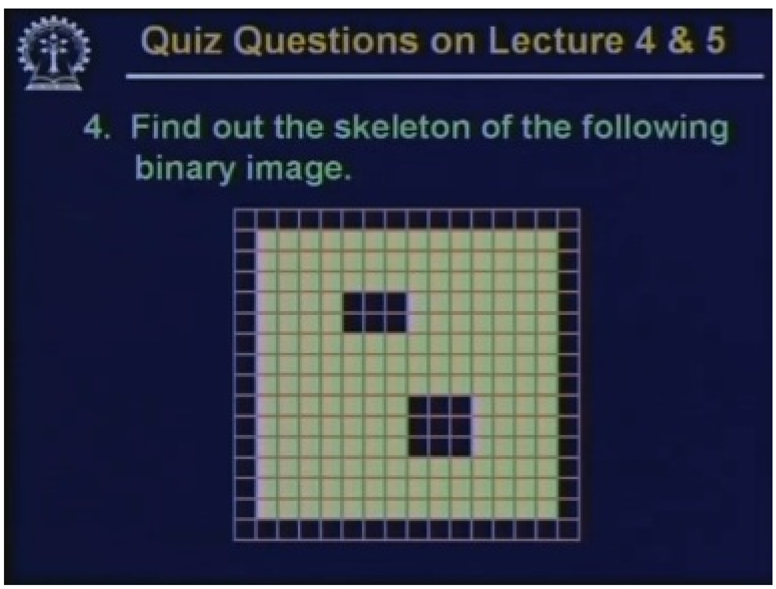
Now, let us give some quiz questions for lecture number 4 and 5. In the first one you have been given 2 points p and q, you have to determine; whether q is a 4 neighbor of p, q is a 8 neighbor of p or q is a diagonal neighbor of p. Second problem, again between 2 points p and q; you have to find out what is the Euclidian distance, what is the city block distance and what is the chess board distance.

(Refer Slide Time: 53:46)



In the third problem, you have been given 2 regions - image region S_1 and image region S_2. You have to find out whether S_1 and S_2 are 4 connected or 8 connected or m connected.

(Refer Slide Time: 54:04)



The fourth problem, it is a binary image; you have to find out what is the skeleton of this binary image.

Thank you.