Introduction to Geographic Information Systems

Dr. Arun K Saraf
Department of Earth Sciences
Indian Institute of Technology, Roorkee

Lecture - 07
Raster data compression techniques

Hello everyone. In this particular lecture, we will be discussing about Raster data compression techniques, and we will be also seeing that, why data compression is required and various types of data compression techniques, which are relevant in AGIS or in digital image processing.

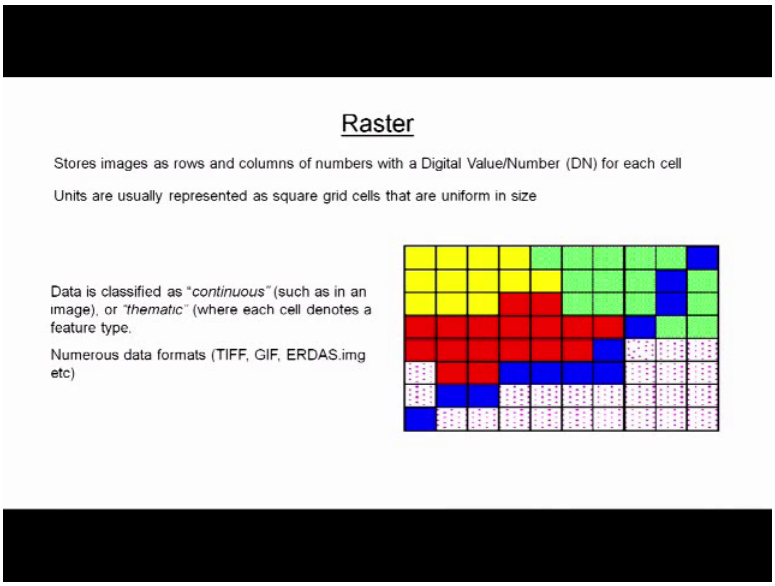
So, we will be discussing all those here. First of all, why data compression is required? As you know that, in, in case of Vector data or in Tin data, there is hardly any redundancy, therefore, data compression has a no role to play basically, but in case of Raster data, for example, like satellite images, or in digital elevation model, might be having lot of redundancy. Redundancy means, for large area, the same value is being repeated and therefore, by some means, we can reduce the requirements of data storage, because Raster occupies large space in hard disc, as compared to Vector data, or Tin data. Those data are very compact data models, but Raster is not in that ways very compact model, and therefore, data compression techniques have been developed, are been developed. Many are in public domain, and many are copy write protected, but still some of them, we can use it.

Ah, 2 major types of a data base techniques, which exist, 1 is a, we can say that, that they are non-destructive Raster data compression techniques, and another 1 is, destructive Raster data compression techniques. Let us take the example of first non-destructive data base Raster data technique, is that, in that ah, whenever you, you are say having a satellite image, you compress it by that particular tool or compression tool, and then, when you want to see the original, you can reach to the original quality, or original size of the image, without ah, compromising on the quality of the image. So, this kind of technique is called non destructive. Example is like with TIF images, you can have a compression, which is called LZW, will see in little later in detail, what is LZW. And using this tool, or technique, you can compress your TIF file, which is a Raster file, or image file.

Now, whenever you want to decompress it, or uncompress it, you can reach to the original. And there are non destructive, or there are destructive image process Raster data compression techniques are, like a, like JPEG. JPEG is a file format, as well as a compression technique. So, if you have converted your file, say, originally it was in TIF you have saved as JPEG, and later on you have deleted the original TIF file. Now you want to restore to, to the original quality you will not achieve. However, there is a trade of JPEG will provide a quite high impression compression, where as LZW in case of TIF which is non-destructive, may not provide that kind of compression. Say many times there is tread of you achieve high compression, and a, but you lose the original quality of the image, or you cannot restore to the original quality. So, these are the complications which are there in case of Raster data compression technique.

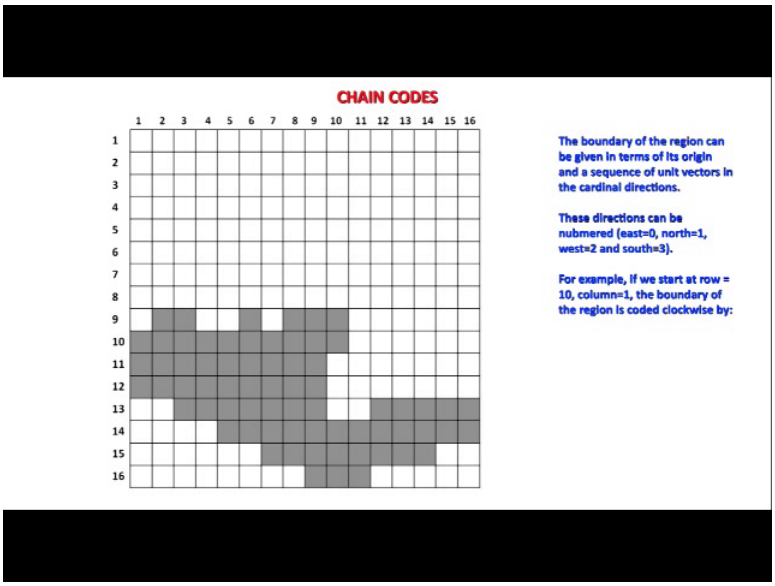
Now, we as we know that, a Raster is stored in rows and columns are a 2 dimensional matrix, that we have already discussed. So, we will not go in this detail, about what is Raster.

(Refer Slide Time: 04:07)



Now let us take the real example which is very simple example, I have taken here is a 16 by 16 binary image, and in one area which is shaded here, say it is black, through it is appearing as grey, just to keep this grid lines appearing, therefore, it is deliberately keep in grey, otherwise, assume that this is a binary image.

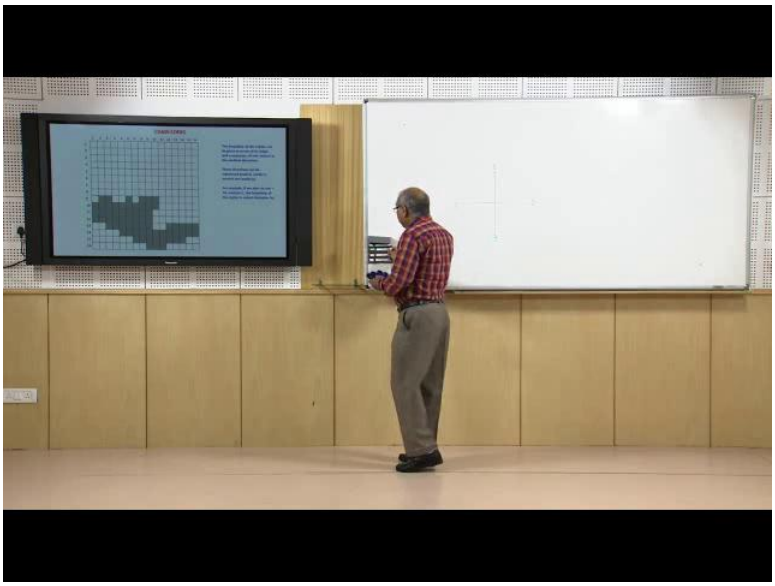
(Refer Slide Time: 04:10)



Some area is homogeneous. There are no other values except the say white, and here the values are black.

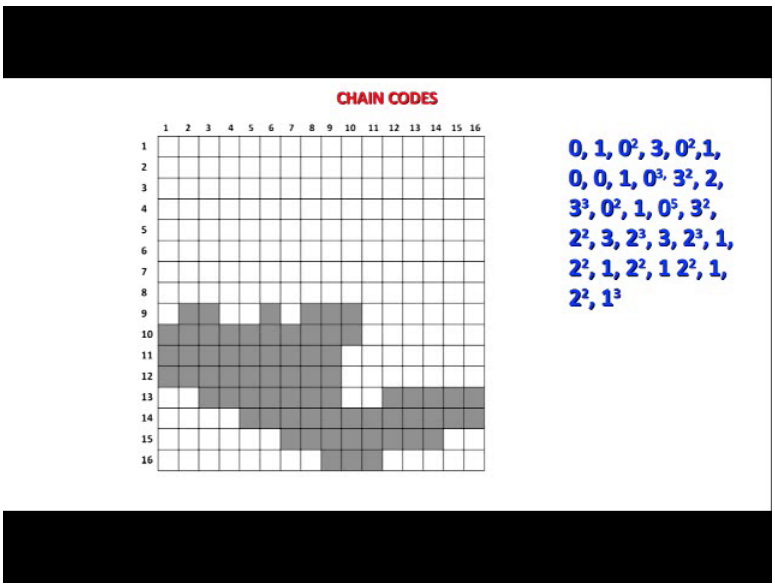
Now if I want to compress this kind of image, then there are 4 basic compression tools, which, which have been implemented to GIS, and image processing softwares, and a every time whenever you go, save as, and some options, you will find these details are available, and sometimes when you are using such software, you do not know why, why such options are available. So, many times now, when now later on, when you will use, or understand all these comp, basic compression techniques, then you would know what is the meaning of those options are.

(Refer Slide Time: 05:37)



So, here I am taking this. So, first I will go for a chain codes. So, if I have to, one type of a data compression technique. So, here what I, I have, I will be following a concept, which is a something like, a that, for, if I move my cursor, into the north direction, I will use 1 value when I move to the east, this is 0 and then to the west, 2, and then for the south, for the south, it is 3. So, likewise, the coding can be done, as per your convenient, but if that that is the coding which is that in here we follow, then this entire shredded area, can be stored quite easily, following these codes.

(Refer Slide Time: 06:21)

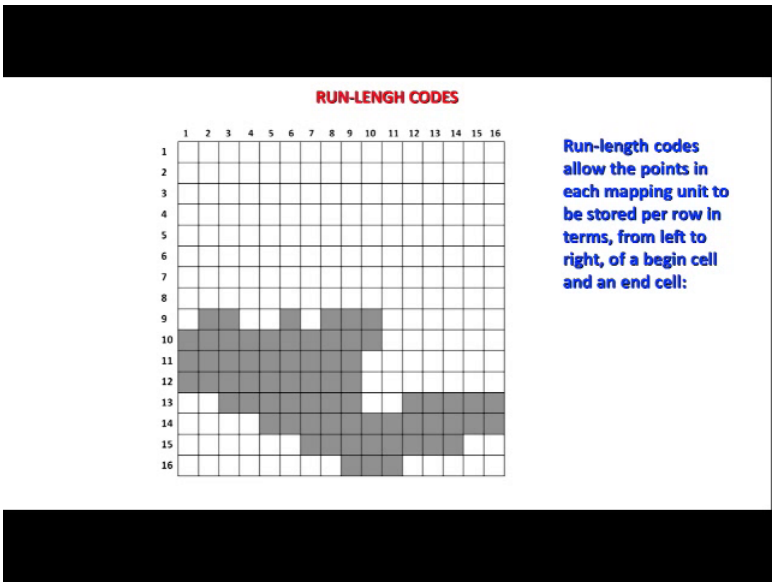


So, we start, because our data is starting in the first column, and tenth row. So, this is, this is stored as that first we move ah, toward the east direction, and this is not arithmetic, these are the codes. So, there is no meaning of that, like you know GY square has, does not having arithmetic meaning. These are just simple codes. So, we move our cursor 1, 1, 1 pixel towards the east, and east has been coded as 0. That is why, 0 is coming first, and then 1 pixel up, and therefore, towards north, we have given 1 code 1, so 1 and then 2 pixels towards the east.

So, it is 0 square, and likewise we come at the end, and then go from here 1 cube. So, we reach to the original position, and this is how this entire image, otherwise it would have been stored in 16 by 16 pixels, can now, using chain codes, can be stored by using just these simple codes. So, because a simple example, a binary image. So, either 1 area, we will store. So, because this black area shredded area is smaller so, it is better to store this 1. So, by doing following those chain code concepts, we can store that entire image, using only few codes.

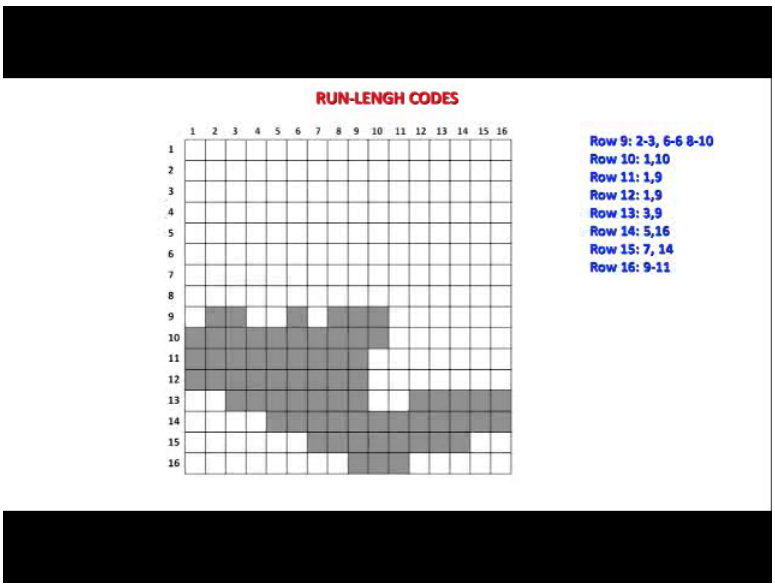
There is another technique, which is called Run Length Encodes, or sometimes it is written as, a in many softwares and elsewhere, it is also mentioned as RLE, E here instead of code, it is stands for encodes. So, that is why E is there. So, Run Length Encodes or codes are there. As name implies, that along the length we run and count the number of pixels, which are coming, over the cells are coming. So, Run Length Codes allow points in each mapping unit to be stored, per row, in terms of left to right, of a begin cell and end cell.

(Refer Slide Time: 08:13)



So, let us, see how this image it can be stored, like here, because up to row eighth, nothing has to be coded.

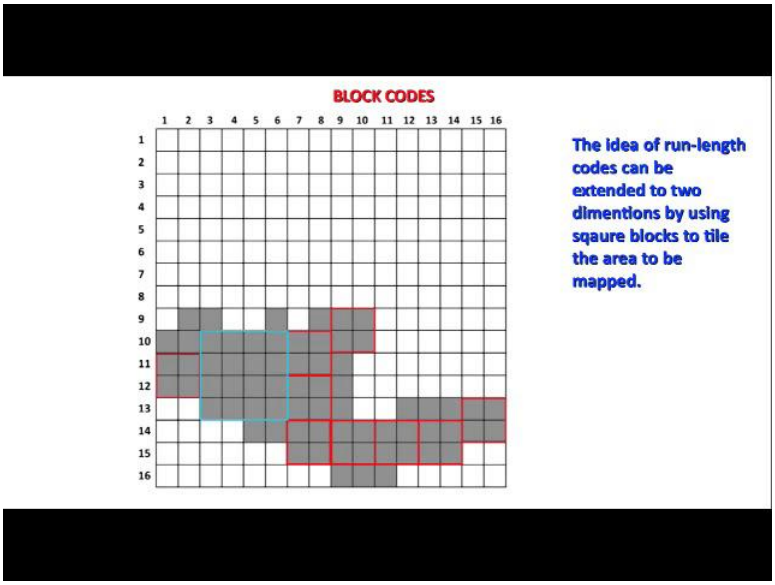
(Refer Slide Time: 06:28)



So, that is not recorded. Now, ah, ninth row we are having first ah, white pixel, and then we are having black. So, we store only starting from black, 2, 2, 3. So, this is length has been stored at row number 9, then row number 9, again the length has been stored, and as only single pixel is there, 6 to 6 only 1 pixel, then there are 3 pixels. So, then 8 to 10.

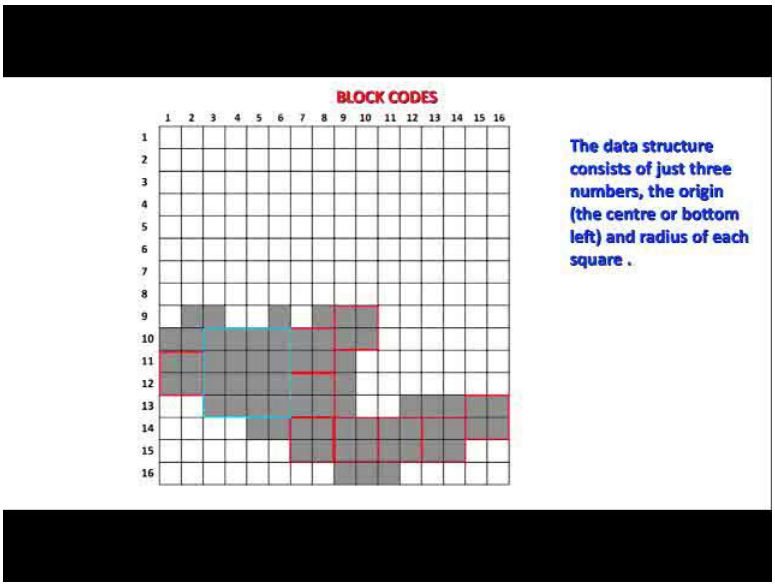
And likewise, for each row, wherever we are having, this black pixel or dark pixel is stored, and a till we reach to the end of the image, the last row in this case, last row column number 9 to 11. So, everything is stored, and first it is searched, and is stored, and therefore, instead of storing again, 16 by 16 image, you store only using these codes. Now the third basic category of a data compression technique, is called black codes, that is the idea of the run length codes can be extended to 2 dimensions, by using this square blocks to tile the area to be mapped.

(Refer Slide Time: 09:40)



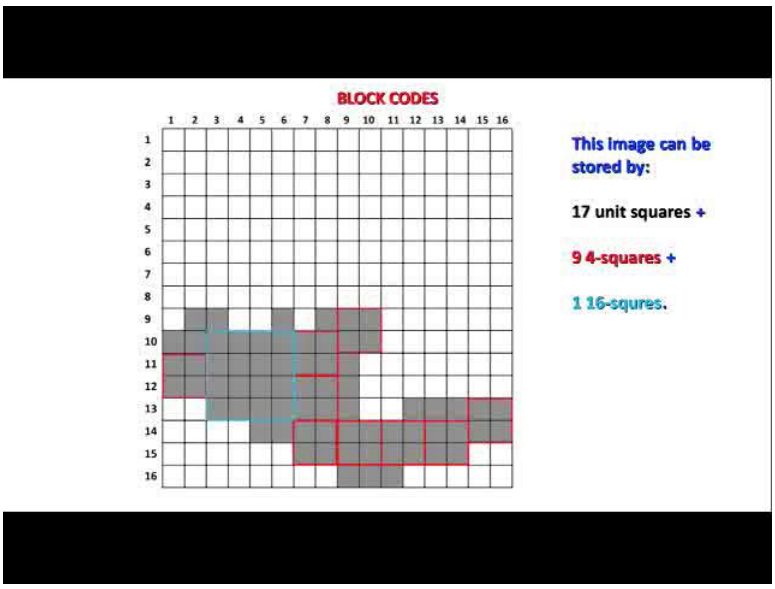
Here, first search of that shredded area is done, largest is squared block has been searched here. In this example, which is shown here, in the blue colour, is the largest - 4 by 4 pixels largest, which is having homogeneity. So, first this is coded, then next smaller, then, 2 by 2 is coded, which is shown in red colour, and then finally, you reach to the 1, or unit level, that is 1 by 1. So, as name implies, first you search the largest 1 block, block by block it is stored.

(Refer Slide Time: 10:29)



And then, if I have to, you know, data structures in black codes consists of just 3 numbers, the origin, the centre, or the bottom left. So, we will store, from where this block is starting, and what is the dimension, or radius of each square, or size of each square.

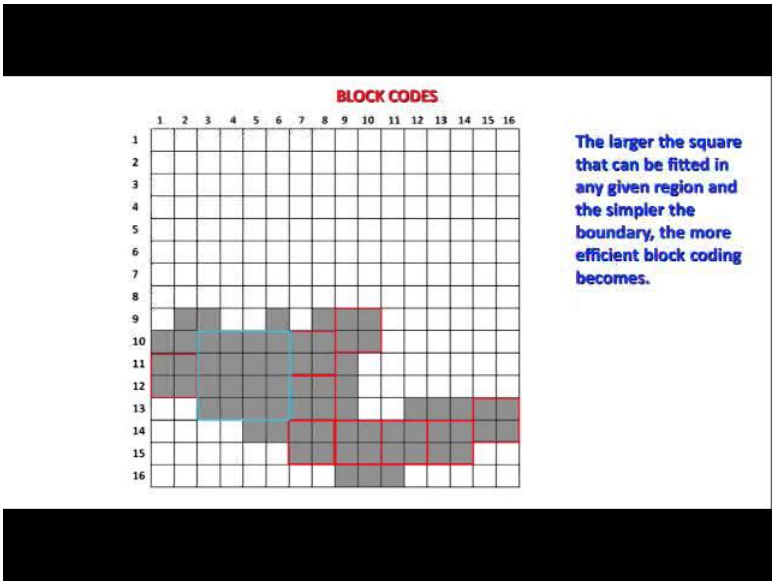
(Refer Slide Time: 10:46)



So, if I have to store, that particular image, the binary image, or 16 by 16 pixels, then only, in this particular example, only 3 things have to be stored, that I am having 1

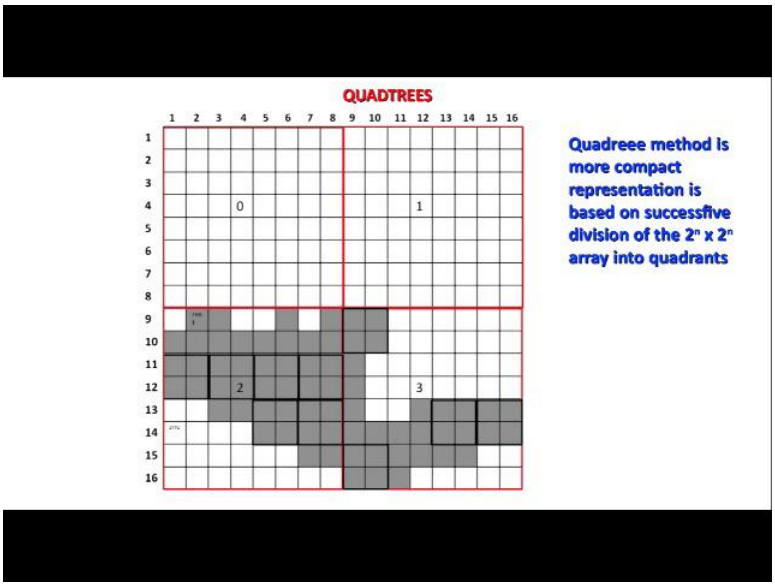
square of 16 pixels, and then, I am having 9 squares of 4 pixels, and I am having 17 squares of unit level; that means, 1 pixel each.

(Refer Slide Time: 11:20)



And therefore, the entire image, can be stored, by using certain codes, And it is quite efficient. Larger, larger the square can be benefited, or fitted in the any given region, and the simpler the boundary, more efficient black board becomes. So, if a your data is having lot of redundancy; that means, homogeneous areas, large homogeneous areas are present in the data, then you will achieve very good compression. And these techniques become much faster, especially about black block codes. Now, the last a in this basic types of, data Raster data compression technique, which is this technique is very popular, and there are various variants of this techniques also exists.

(Refer Slide Time: 11:59)



Like instead of quadtrees, people also have developed oct trees and so and so forth. Here, this technique, initially, it is not simple, but it provides quite good compression. So, what it is done in this technique is, the first, this entire image is searched, in this example, very carefully I have taken the binary image first, and in real, in real scenarios, you will be generally you may not be having binary image. You might be having coloured image, and 24 bits image and 64 bits image. So, the complications will be much larger, 1, and second, in this particular example, I have taken a squared image, and therefore, to demonstrate the quadrics concept, it is easier. But when your image is not a square, what would happen, that is scenario we will also see, but anyway.

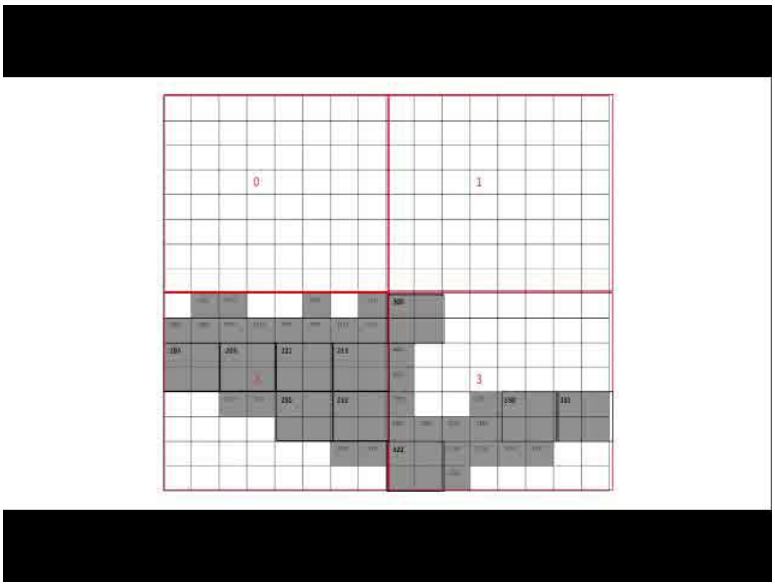
So, lets start with a coding this 1 through using the concept of quadtrees. So, first the search is made, and then 4 quadrants, the entire image are divided into 4 quadrants, and then each quadrant is further searched. If homogeneity has been achieved, within that quadrant, then no further divisions would be made. Like in this example, the quadrant number 0, now there is no heterogeneity is there, everything is homogeneous, all pixels are having same value, and therefore, no further divisions 4 quadrant 0 is required, same in case of quadrant 1.

Let us take example of quadrant 2, in case of quadrant 2, there is heterogeneity exists, within this quadrant, and therefore, more divisions would be required. So, first again this, quadrant 2 is divided into 4 parts, then again search is made. So, if we follow this 0, 1, 2,

3 code system, then again this is 1, and we will code as 2 1, 2 0 sorry, 2 0, 2 1 and a 2 2 and 2 3. So, now, this quadrant 2 2, has is a - is still having some heterogeneity, and therefore, there will be further such divisions, till we reach to the unit level. And as we can read here, is that quadtree method is more compact representation, and therefore, it provides high compression, compares to other discussed compression techniques, based on successive divisions of 2 power n by 2 power n array into different quadrants.

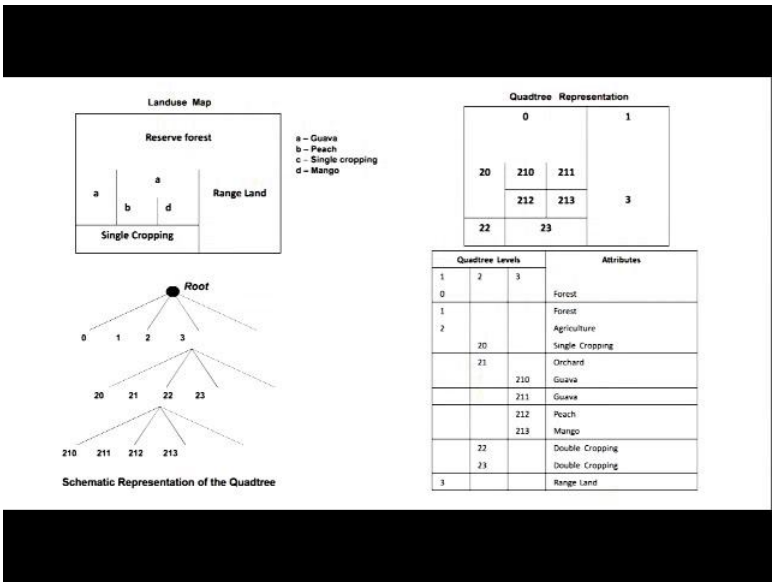
And if I take the example here of quadrant 0, then how, how for each pixel is coded. So, this is the largest, largest quadrant 0, 1, 2, 3. Within this then, divisions have been made, and this is the 0. So, it becomes 2 0, now 2 0 is further divided into 4 quadrants. So, this is 2, 0, 0, and then this 2 a 2 0, 2 0, 1 0 and so and so forth. But less concentred on this one, this is, this is 2, 0, 0, 1. So, likewise for each pixel, a code is given, and whereas, when we do not have any divisions or homogeneity is there, no further codes are assigned and no further divisions are made, and therefore, it is very compressed method and provides good result.

(Refer Slide Time: 15:52)



Less, the same, same thing I have been, I have coded here, for different pixels, and as you can see this is how it is seen.

(Refer Slide Time: 16:03)



Now, why it is called Quadtree? Because it is inverted tree basically, concept wise. So, you are having one route, one image, and then it is divided into 4 sections and 4 quadrants, and then homogeneity is searched. Homogeneity is achieved then, go for next, and go for next, and likewise, till you reach to the unit level. ln the earlier example, I have taken image with is 16 by 16 pixel; That means, a squad image, but if you recall that, when I, when we were discussing about Raster data, the overall shape of the Raster can be either square or rectangle; However, the unit of the Raster has to be in square in shape. So, if we, in the previous example, I have taken Raster, or an image binary image of square shape.

Now, let us take example of real map, which is not in a square shape which is in a rectangular shape. So, what would happen in case Quadtree? So, this is how what would, what it is done, before, before you know, the quadrant are searched, first it is searched, whether the input image is rectangular or a square. If it is rectangular, then on the sides, few rows, blank rows are added, columns are added, and the first the entire image is made in square form, and once it is in square shape, then divisions will start, like here it has happened. So, the quadrant 0 is, is going beyond the boundary, which are shown here.

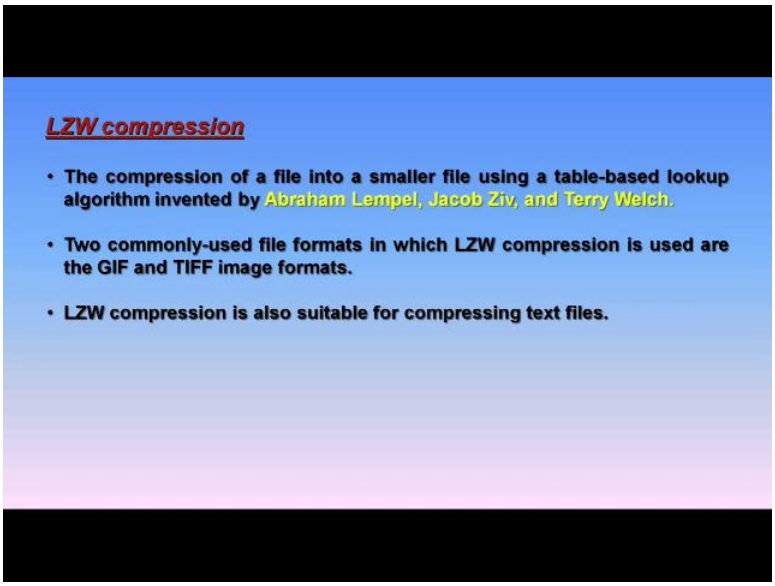
So, you have to imagine that, quadrant 0, is much larger then what is shown here. And it has to be square in shape. It cannot be rectangular. Same with the one, because it is the

quadrant 3 is not sufficient here. So, it has go on the right side further and likewise, but if I have to code this map, using quadtree concept, that this is how it is need to be done, same scheme we follow 0 1 2 3. Now the quadrant 2 has been further divided, and the boundary is here, between 4 quad basic quadrants. So, it is 2 0 it is 2 1 2 2 and 2 3. Now 2 0 2 2 and 2 3 do not require any further divisions, because homogeneity has been achieved. Only 2 1 requires further division. So, then again further divisions, 4 quadrant we have created 2 1 0, 2 1 1, 2 1 2 and 2 3, and we have reached to the unit level. Now no search, no further divisions are required, and by which, you can achieve, the high compaction of the images.

And this is how, if we want to represent a form of tree, then this is how we should be represented. This is 0 1 2 3 3 should have come here, 2, 0, 2, 1, 2, 2 and 2, 3 have should have come here, and likewise, and the in the tabular form if we want to understand, this is how the coding is done. So, an image whether is square in shape, or rectangular, still it can be coded, using Quadtree. In case of oct tree, instead of 4 divisions you make 8 divisions, but the concept wise it remains same, they are all hardly much differences in the result.

Now, as I have said that the very popular ah, your image formats like TIF, or some other image formats, are also supported with some compression techniques, and one of the very popular one, which you would find these options, whenever you save your image, or save as, you go for another format, then sometimes these options are available, and that, do, would you like to save your image with LZW compression. So, what is LZW compression? This compression is of course, the purpose of all compression techniques, to reduce the size file size.

(Refer Slide Time: 20:03)

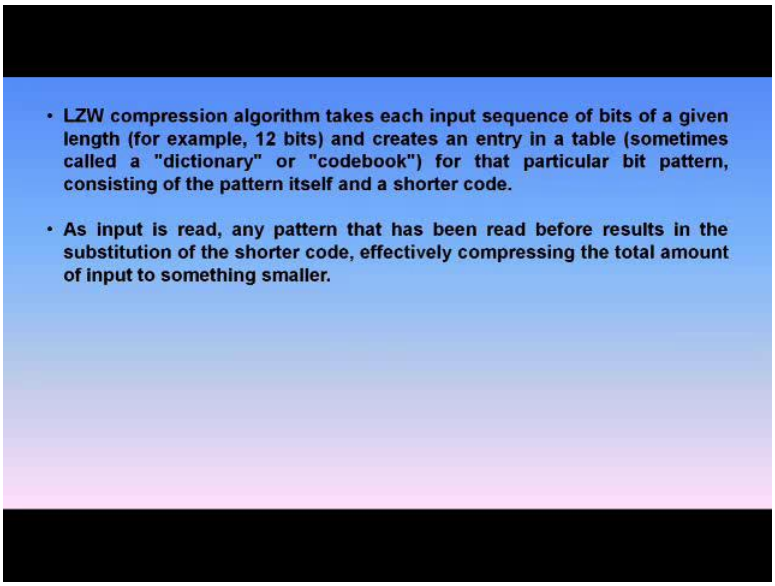


But the same time you have choose whether the, later on if I want to reach to, to the original size, and quality, I should be able to do it or not, that, that is very much important, that is very much required.

Now, LZW is based on the 3 ah, mathematicians who developed the algorithm and their surnames', first character has been taken, Abraham, Lempel, Jacob Ziv and Terry Wetch. So, these 3 people, their surnames have been taken, and LZW. So, they developed, basically the algorithm, or well, which is became very popular, to compress your GIF images, and TIF images. GIF images are mainly used for animations, I will show you little later, some examples of GIF images, and how they can be, how they can be compressed using LZW and, how animations can be created.

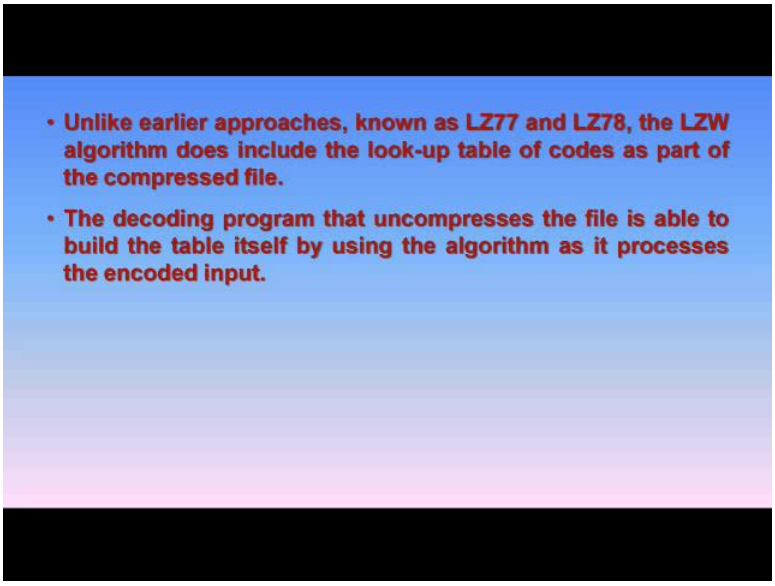
Further that led also suitable for compressing text files, generally text files do not have much redundancy, except when we type lot of space, in the files, that may create lot of redundancy, in even in text file. So, LZW can also be applied there, but most of the time compression techniques, we go mainly for Raster data compression; however, there are many other popular compression tool available, which we use in our day to day usage of computer like winzip or rarzip. They compress, each and every type of file, but all details how they compress are not available to us.

(Refer Slide Time: 21:59)



So, there is beyond discussion of this lecture. Now LZW compression takes a each input sequence, to a bits of given length, for example, 12 bits creates an entry in a table, and then dictionary, then code book, because everything is coded, and the as in other examples we have seen, and then we read this thing, and this is the LZW is non destructive; that means, you can restore to the original file size, and quality, without a compromising on the quality.

(Refer Slide Time: 20:30)



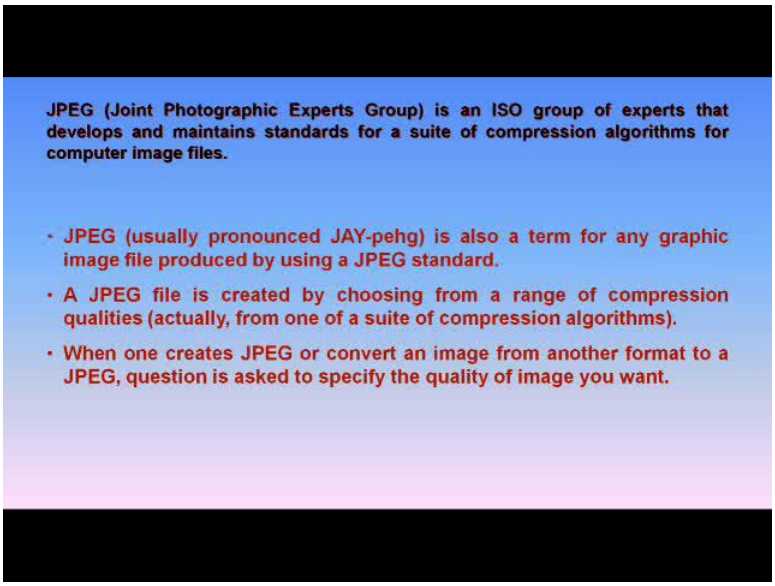
Unlike, a, unlike earlier approaches, there are different versions of LZW, like now a days different versions of JPEG files are also available. So, they, they are further improving on the compression, and sometimes what happen, that if you have compressed with simple LZW, and a later on you are using some other tool, you may have difficulty. So, it is a again trade, of if you go further lowest compression option, at least you will not have problem on others system. Or, if you give the file to somebody, he might not be having problem. So, these are the things before compression, one has to think. Decoding programming that uncompress the file is able to build the table itself by using the algorithm, that is possesses the encode input. All these LZW compressions have been implemented in all well known GIS and image processing softwares. Most popular one is JPEG, and unknowingly people are using JPEG.

Unknowingly means, they do not know, that this, though it is image file format, but it is a compressional tool, data compressional tool. And what it does, it destroys the original image, and quality as well, and you can compare, by, if you zoom TIF file, and the same file, if you save as JPEG, you zoom side by side, of the same level you would find the quality has deteriorated, but it is a trade off. That means, a compression, in through JPEG will provide high compression, a 10 MB file may be reduced to 1 MB file, depending on file to file, because if redundancy is high, then you will achieve more compression.

But the disadvantage is that it deteriorates the quality. So, it is providing you high compression, but the same time it is deteriorating the quality. And lot of people, unknowingly are using JPEG format to store their original files, or original images, original photographs, when take digital photographs. So, they are going for higher and higher spatial resolution, digital cameras, even in their mobiles or otherwise, and that the original files they are restoring, on in JPEG.

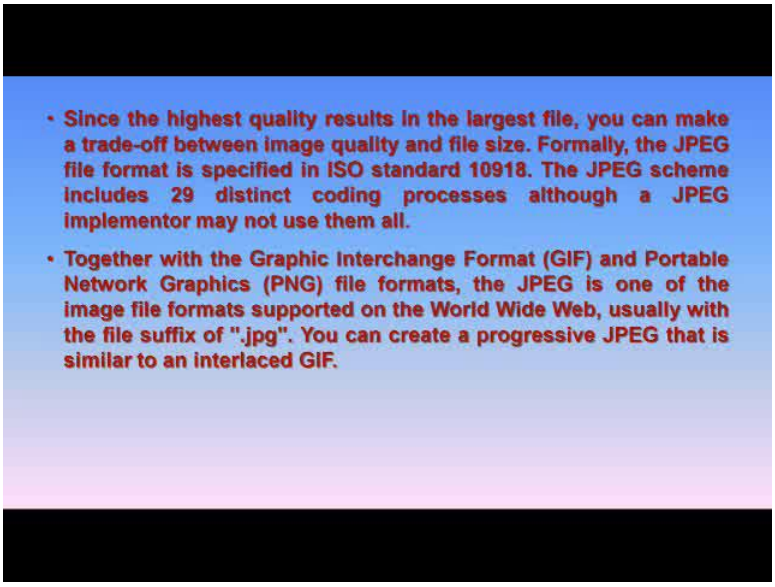
Though they are restoring more images, but at the same time, they are reducing the quality of their images. So, there is again trade off that rather then storing files in JPEG, originally one should store in TIF, and later on you can compress. So, you will have 2 sets of files. One is in TIF, and one is in JPEG. JPEG will provide high compression, but TIF will have the high quality. So, unknowingly, that is why I have said, the people are storing data in JPEG, without knowing, that they are loosing the quality of their photographs.

(Refer Slide Time: 25:35)



JPEG again is a, is a group of people, which is a joint photography experts group. Different versions of JPEG have also come, and a these still are been developed, and because it provides high compression and therefore, it is very, very popular among people. People have followed certain standards, and other coding is also available.

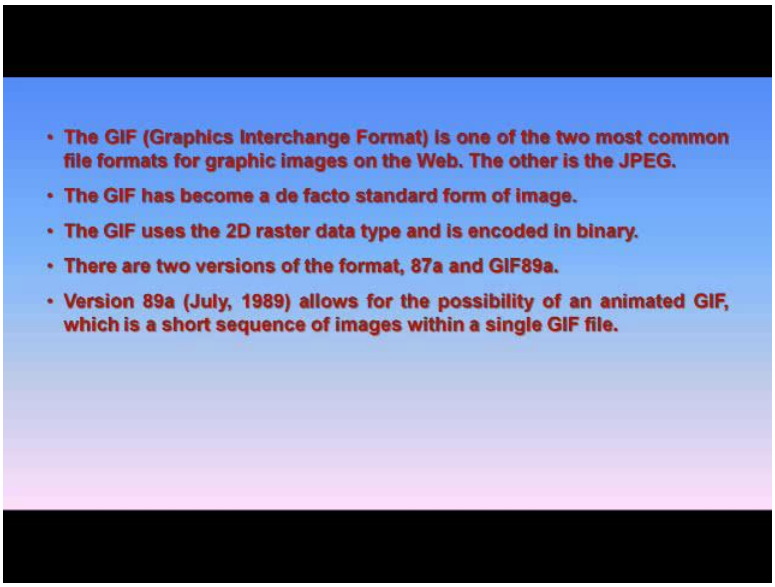
(Refer Slide Time: 26:01)



Then, you know there are, together with JPEG, and they say that your GIF and PNG, and that the portable network graphics file, the JPEG is one of the image file format supported on the World Wide Web. Because on net it is very popular, that is why people

are going GIF for animations which I will show you little later. Now GIF is a graphic interchange format, and here what you can do, you can keep lot of frames in one, like a movie and then you put in a one file, by using certain tools GIF create us kind of tools.

(Refer Slide Time: 26:44)



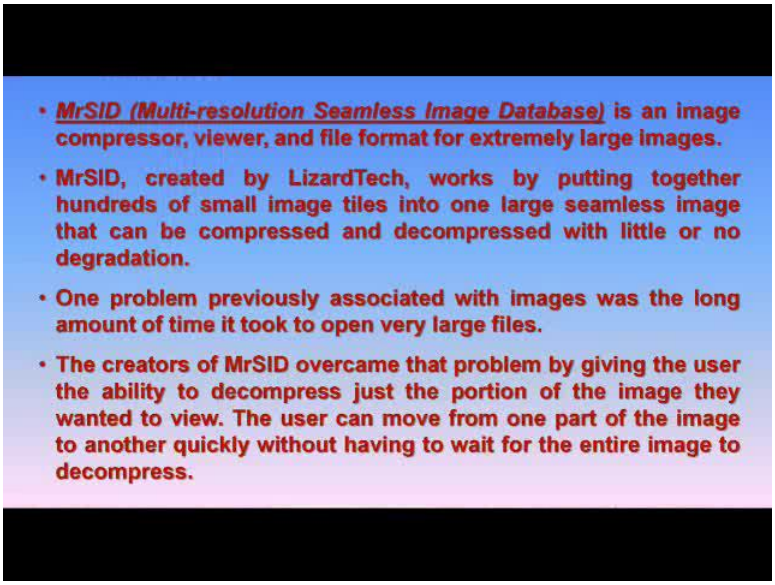
And then GIF file is created, that is also a compressed format. And it works very well on, on any computers. And it is very easy ah, on internet as well, that is why lot of animated images on internet you see, generally they are in the GIF format.

(Refer Slide Time: 27:03)

And the example which I am going to show here, it is having 8 frames in, in a 1 GIF file, using a, a tool, GIF create a kind of tool I have used, and all input images had the same sizes, except that files inside are different, and it is showing a sort of simulated animation, and a you know, the position of sun is moving, at different locations, and the same terrain is looking differently.

The purpose here is just to show the, GIF file, but not the, what basically it is showing. That is a discussion of other lectures. Another example of GIF is, here the sun is coming from almost 5 degree above the horizon, and going overhead, and moving in a loop. So, 1 single file, which is in GIF format, can have animations, and that is the advantages of having animated files in GIF format. And you can put on your webpage, or you can give to somebody, and when they will open it, it will run exactly in the same way. Now there are different versions. So, GIFS are available, further improved versions are becoming available.

(Refer Slide Time: 28:32)

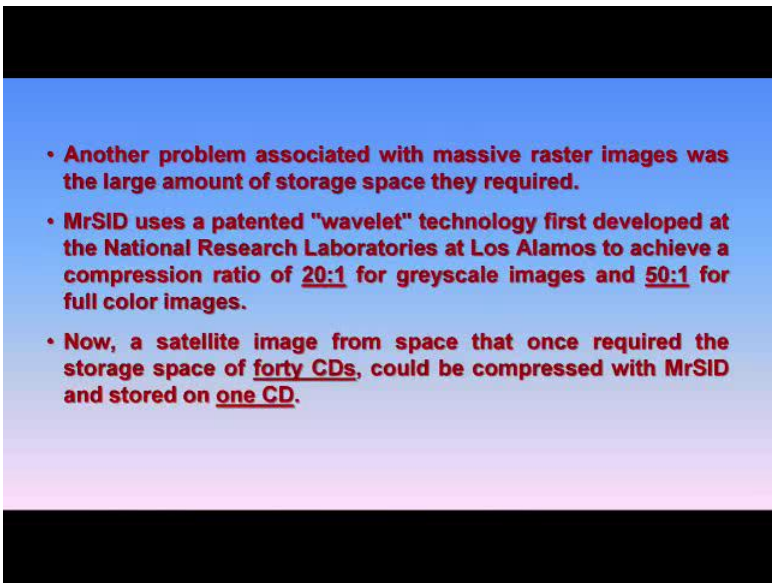


So it is quite good, quite popular, for animations. Now a one more a, a, compression techniques, which has become very, very popular now a days, because it provides very high compression, one, and it is non-destructed image compression technique, and that technique which is called Multi Resolution Seamless Image data base, or MrSID in short we call.

Ah, this has been developed by company which is called Lizardtech. The problem with this that, we do not know much detail about MrSID - this is copy right protected, many softwares had been implemented, but up to only certain limit, for example, like in addas image processing software, or in RGL software, you can compress a file of only up to 500 MB, if your input file is more than 500 MB, then you cannot compress using this MrSID, or option of MrSID, you have to buy then, the permission or excess through the company which is Lizardtech. So, it is a copy right protected. The, what kind of compression it provides is very high, unparallel by any other technique, because the technique which they have developed, is a based on (Refer Time: 29:51) and it provides about 50 times; that means, if 50 m b file, can be compressed to 1 MB, and then, it can, through net, it can be transported, or transmitted from one end to another, very easily.

One of the examples, where MrSID a format is being used extensively, in like Google earth, you know that Google earth is having large sizes of image, of a you know, of many terabytes, but on our machine, on our desktop, this comes very easily, and why it is so, it is easy to see, on our image, on our screens, because they are in the background, it is it has been compressed, and when you open in your machine, it, it is decompressed very quickly. And since it is not destructive, so, whatever it has been stored inside, the system, same you are able to see on your system, as you keep zooming, different layer will come and so and so forth. So, that provides a very compression of about even up to 50 times.

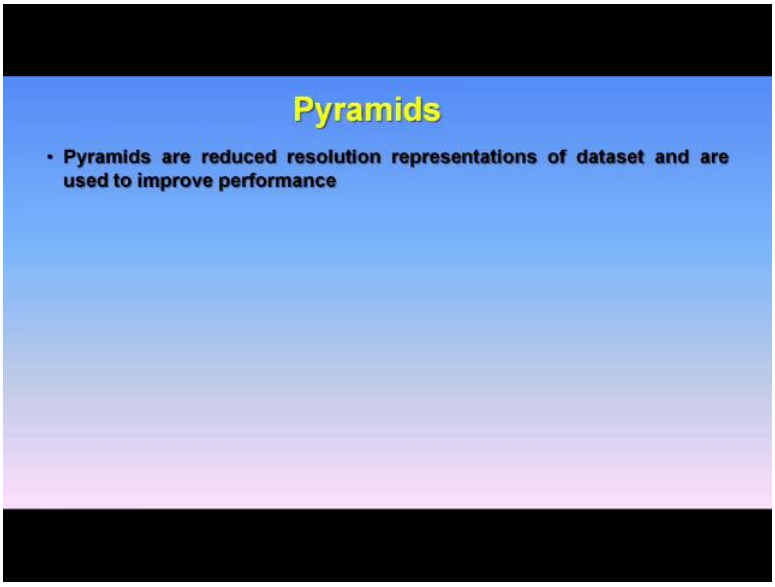
(Refer Slide Time: 30:55)



So, a 50 MB image can be reduced to 1 MB, without losing image quality. That means, you can, this is non destructive image compression technique, or Raster compression technique, and you can restore to its full quality, anytime whenever you want. So, that is the biggest advantage, but it is based on, it is copy right protected therefore, we do not know much detail, only thing that we know, it is, it has used the wavelet technology. One example is also given, that if you are storing a satellite images or in 40 CDs, then entire, those images in MrSID format, can be stored in 1 CD.

Now the last thing which I want to discuss in this data compression techniques, if this is not exactly data compression techniques, but it provides a speedy display of your certain images, or Raster, on GIS platforms, or an image processing softwares. This is a concept of pyramids, which is also followed in Google earth, that, what is done here, that it reduces the resolution, at the highest level.

(Refer Slide Time: 32:08)



So, when you are seeing the entire globe, you are seeing at very low resolution. As soon as you start zoom in, you get the higher resolution image, and therefore, there is a construction of pyramid.

In a, in these like examples of the software like RGIS, or a dos image in software, whenever you open a large file, it will ask you the this option, that would you like to construct a pyramid. And if you say yes, then what it will do, it will create 2 or 3 files on your computer. And that those files will not be very large, very small space occupying

files, and they will store this pyramid structure. So, next time whenever you open that file, it will not ask whether you would like to construct the pyramid, because it has already constructed one, and your display becomes much, much faster. And this is the technique, in, that is why I have put in the compressional techniques, but exactly it is not compressional technique, it is it makes your display very efficient and quite fast.

So, the pyramid has become very popular structure for representation, whether it is Google earth, or in your systems also, you can opt for this only once, first time you have to create pyramids, rest of the time, no pyramids are required. If you transport the file from one system to another, then the original file, plus these 3 files have to transport together. Then you need not to again construct the pyramid, and constructing the pyramid of a large image would not take much time. So, it is very efficient way of a displaying new data, on your screen. And as you go high, as you keep zooming, you get the fine resolution data, otherwise at the top of the pyramid; at the initial stage you get very low resolution data, from the same image. So, that is the advantage.

Thank you very much.