Computer Graphics
Prof. Sukhendu Das
Dept. of Computer Science and Engineering
Indian Institute of Technology, Madras
Lecture - 43
Digital Image Processing

Welcome back to the last part of the lecture series in computer graphics. In the last class we had discussed aspects about digital image processing mainly the concept of digital image compression the jpeg format or jpeg. Towards the end of the last class we just covered in very brief the concept of digital image enhancement and we also looked at what is contrast stretching. So we will start from that were we left-off and we will discuss concepts of digital image enhancement. This is an important aspect of course in digital image processing or digital image manipulation where you have digital images stored in a certain format it could be JPEG TIFF BMP or whatever it is.

And then if the quality of the image which you have obtained is not very good in the sense that it could be corrupted by noise or due to the sensor effects that the illumination is not good or well distributed all over the image you would like to enhance the image quality.

So we will talk about concepts of digital image enhancement and there are many ways by which you can enhance a digital image. And it will be covered in the extensive course on digital image processing and just to name a few there are two broad categories of digital image enhancement techniques. One based on specter or Fourier based filtering techniques as they are called or based on digital Fourier transforms which we will not cover here. The other category is based on spatial transforms or transformation done on the image pixels itself in the spatial domain.

So we will mainly concentrate on spatial domain analysis and that too we will talk mainly about three different types of digital image enhancement. One based on contrast stretching the other is probabilistic distributions which are based on the image histogram. So we will understand what an image histogram is and we will talk of the simple technique of histogram equalization to enhance the image. And of course the third category we will also see the concept of filtering an image, the last part. So let us look into the slide of digital image enhancement. That is the topic under digital image processing which we will be talking about today.

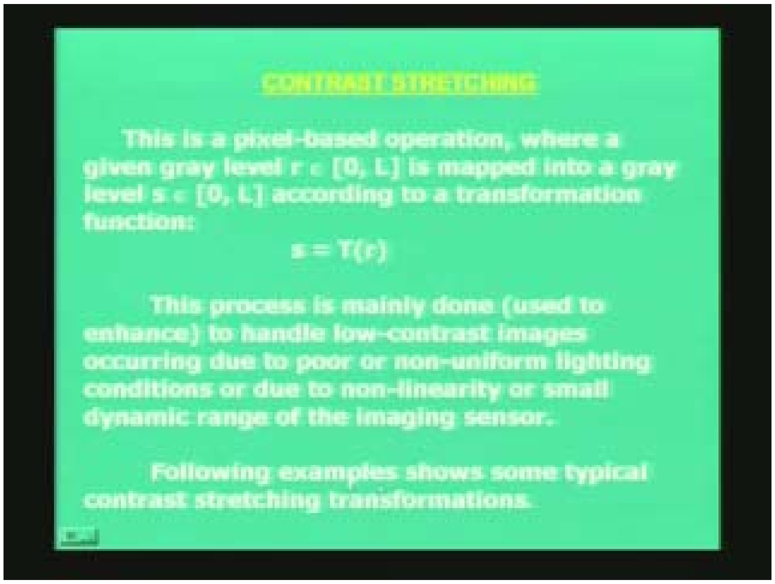
Under digital image enhancement the first concept is contrast stretching. this is a pixel based operation where a given gray level within the range 0 to l this value of l depends on the number of bits used to store a gray level value or a pixel value and there is a variable r lying within this range and that is mapped onto another gray level s which also lies in the same range 0 to l according to a transformation function s is equal to t of r.

So depending upon s equals t of r you choose you can have different type of enhancements. In fact you can have de enhancements also. you can degrade the quality of the image, you can also increase the quality of the image and enhance it or de enhance it that means attenuate in some sense those gray level values depending upon your requirement.

So s is equal to t of r, r is the variable used for the original gray level value or pixels in the image and the s is the variable for the transformed values. You are transforming values from r to s using the transformation function t of r. That is what we have seen here and what is the main purpose? The process is mainly done and it basically means it is used to enhance images and it is mainly done to handle low contrast images occurring due to poor or non-uniform lighting conditions or due to nonlinearity or small dynamic range of the imaging sensor.

So the image could be degraded due to various reasons. There could be nonlinearity in the imaging sensing device, the dynamic range of the response of the imaging sensors on light falls on the sensors the light energy is transformed to electrical energy then it is converted to a digital value and that is how typically the CCT based sensors or even any other electro magnetic sensors or solar sensors work.

(Refer Slide Time: 06:10)



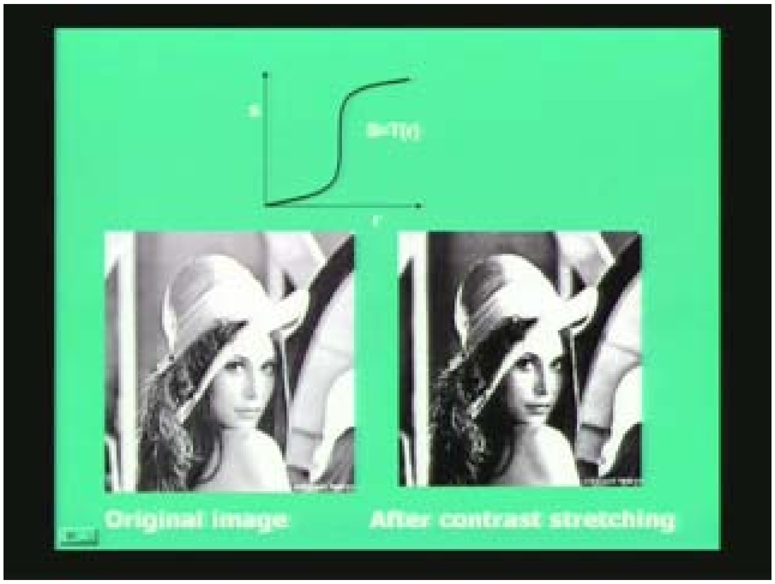
Those sensors may be faulty or their performance range or dynamic range of output values is not large or wide enough for you to give a high contrast image or a good quality image.

So in those conditions you apply a transformation s is equal to t of r. and the last part is we will see examples, the following examples will show a typical contrast stretching transformation. We will go through a few examples of very simple contrast stretching transformations which we will see now.

Let us take this particular example. As you can see we have also used this example in the last class to demonstrate the concept of image compression using jpeg format, the original image of Lena and we use a transformation function of nonlinear function of this nature s equals t of r s equals t of r so r is being transformed to s using this function. Basically what you mean is you pick up a pixel value which could be bright or dark depending upon where you are.

You can actually do row wise or column wise scanning and for at any particular point x, y and the image you pick up the grey level value go to that in the horizontal scale in this s equal to t r function what is the value of r and corresponding to that find out the value of s. So if it is higher you will be getting a higher value of s if it is lower value of r you will get a lower value of s.

(Refer Slide Time: 06:22)



Now if you see what is the purpose of this type of or nature of this transformation function, what it basically does is low values of r are made still lower pushed down and higher values of r are pushed to larger values. And that results into an output image of this nature here as you can see the brighter portions have become much bright and darker regions have become very dark.

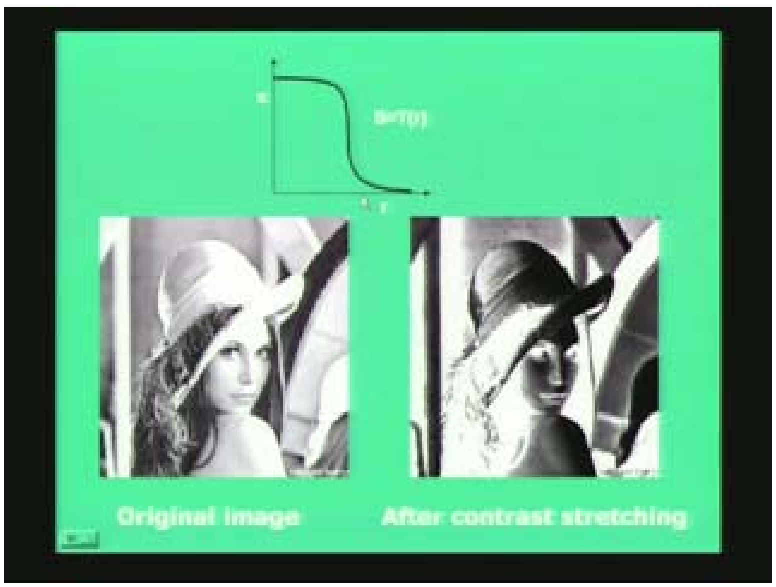
You can almost guess that if s equal to t r function is a linear function where s is equal to r with the slope is equal to 1 the output image after contrast stretching will be the same image as the original image if s is equal r.

However, if you use such a transformation function where darker values are made much more darker brighter values are pushed to the large brighter range this is what will happen where you see dark regions will become still darker and brighter regions will become very bright. So, this is an example of contrast stretching here s equal to t r.

Let us take just the inverse effect of this transformation function and see what is the effect. Go to the next slide, as you can see here if you take s equals to t r of this nature where darker values of r in the original grey level image are transformed to a high value of s and brighter or large values of r transformed to low values of s this is the output which you will have after contrast stretching. What happens here? As you can see brighter values here or on the top or on the shoulder have all become very dark and darker pixels as you can see here which is this patch or regions in her hair here have all become very bright.

So it is just equivalent to saying that you have actually got a negative of the image in some sense. Of course a perfect negative image can be obtained when the slope of this is linear and s will be a constant minus r then you would have a perfect negative image but this is almost negative like.

(Refer Slide Time: 09:19)



You do have very dark pixels and very bright pixels just the opposite function. If you just look back and compare these two transformations functions and the corresponding output let us roll back and see this what was we showed earlier s equal to t r brighter pixels become more bright, darker pixels become very dark here or even you can see the reverse effect is here as you can see that dark pixels become very bright and bright pixels become very dark here.

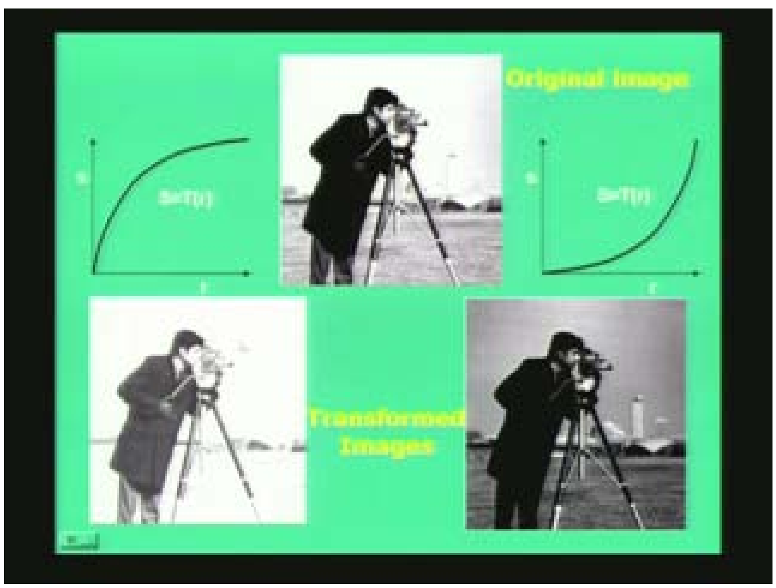
This is an example of contrast stretching in either direction. You are free to write a small program and take any arbitrary s equal to t r function and observe the effect. You can actually have magical effects both in a grey level and a color image you can experiment for yourself and see the result of this contrast stretching operation.

Now let us take another example, as you see here the original image is given here and we expose it to a transformation function of this nature s equal to t r. Now this is also a nonlinear function. It is basically an exponential curve in the sense that of course is something like a Bh characteristics in electromagnetism. And if you provide such a transformation function where you pick up pixel values from this image find out where it is in the r axis corresponding to that find out what is s axis and if you do it for all the pixels in this image what will be the result? You will get something like this, as you can see the image has not changed much in its quality but you can see the background.

You can see the background here they have almost become very bright. If your TV quality is good I mean the media in which you are observing these as you can see here this is an example of what is called a gamma correction in image processing or even computer graphics.

This is an example of a gamma correction. I will avoid the equations but you can implement this gamma correction using this transformation function as an example given here. And the gamma is the parameter of the curve, in this case gamma is less than 1, the lesser the value the more steeper the curve will be. In fact gamma is equal to 1 the s equal to t r function is s equal to r example that means a linear one from left to right as you can see here.

(Refer Slide Time: 12:12)



This is an example of a gamma which is less than 1 more lesser than 1 will be more sharper when gamma is equal to 1 you have s equal to t r. When gamma is more than 1 this is what will be the result of the curve and the corresponding output as you can see here. This is definitely much different than the original. As you can see it is much darker with this transformation function for gamma greater than 1 here and this output using this transformation function you have this output whereas from the original image using this transformation function you have the transform. These are the transformed images obtained by the corresponding s equal to t r transformation functions. These are certain examples of contrast stretching and by s equal to t r function. Now what this s equal to t r function does is it does not worry or tell you what the image is in terms of in what sense it is poor.

You are blindly applying s equal to t r function on a darker image or on a brighter image or you can apply it on a richer contrast image already and in some cases you may land up with poorer results. It is not automatic or in some sense it does not derive out any information from the image and then adjust the parameters of the curve s equal to t r so it does not do that.

We need certain enhancement techniques automatically in certain digital image processing applications where the image should be, if at all if the contrast is poor in the sense that it is very dark are things could be very bright in either sense the images are bad. If the pixels mostly are very dark or very bright the images in general are not often of good quality. When you say subjectively of course that the images are of good quality you typically have all dynamic ranges of pixels starting from some pixels which are very dark to some pixels which are very bright.

We will see with an example later on but the concept of or the concept which we are going to discuss right now talks of an automatic or semi adaptive method by which the information is derived from the image about whether in what way the quality is poor and then adjust the pixel values using some transformation.

Adjust the pixel values based on certain transformation and we are going to study the concept of what is called the Histogram Equalization based technique for image contrast enhancement. That is the next slide now. Often we will use in short HEQ term which will mean nothing but Histogram Equalization.

(Refer Slide Time: 14:35)

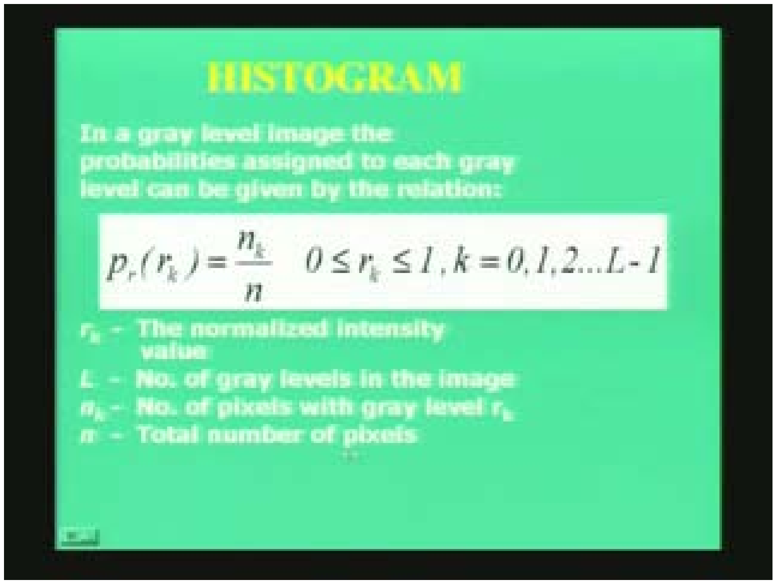
Before going into the concept or algorithm on Histogram Equalization we must first understand what a histogram of an image is. So what is the histogram of an image? Let us look back into the slide and try to understand. In a grey level image the probabilities assigned to each grey level can be given by the following relation.

(Refer Slide Time: 15:07)

Let us introduce these variables and then we will look into the expression which I am not bringing right now. So we are talking of the probabilities assigned to each grey level and where rk, so you remember this r variable was there, now is that the k subscript has come which talks of discretized values of r the normalized intensity value rk and l be the number of grey levels and this variable is also not new to us. These two are new variables nk is the total number of pixels or the number of pixels with grey level value rk.

Now rk could run from 1, 2, 3 and so on. So for a certain value of k where k is an integer and we have a grey level value rk and for that the image has nk number of pixels that is the most important part. Basically you have to count the number of pixels you have in the image which have the grey level value rk so that is the value nk here. So as you can see here nk being the number of pixels with grey level value rk and of course small n is the total number of pixels.

(Refer Slide Time: 16:44)

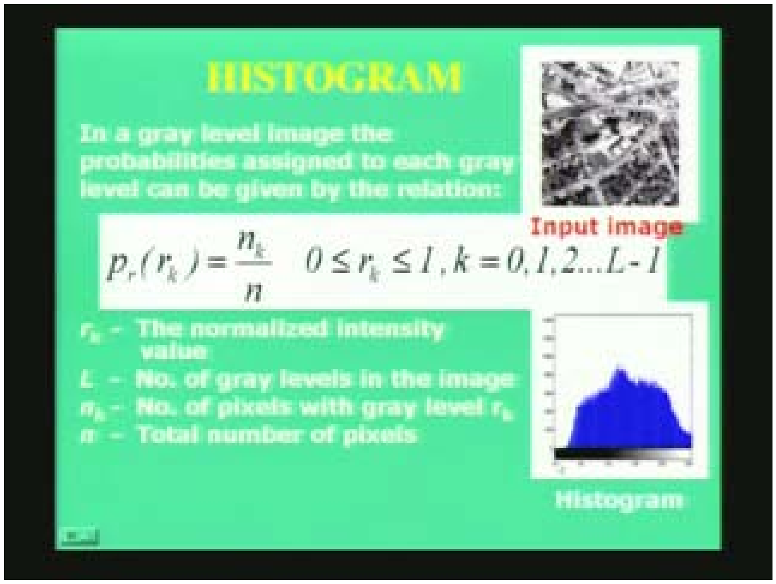


The formation of all these nk will be n as the total number of pixels in the entire image.

If I have understood these four variables here this is the expression of what we mean by the probabilities assigned to each gray level value. pr of rk is given by nk by n. You know what is this nk? nk is the number of pixels with gray level val value rk and the denominator is the n which is the total number of pixels. So this probability value will be less than 1 and it will be running between 0 to 1, nk maximum can be 1 at the most and the rk values we will say normalized intensity value so it runs in the range 0 to 1 remember k is an index so it must be integer running from 0, 1, 2 l minus 1. It is because if l is the number of gray levels in the image as given here k runs from 0 to l minus 1.

So there are l different levels and the level runs in the normalization 0 to 1. It is just a normalized range, these are fractional values rk and for a certain value rk given an index k nk is the number of pixels with that corresponding value rk divided by the total number will give you the probability that so many number of pixels will have a value rk. So they plot of this pr of rk versus rk is actually the histogram of the image. So let us look at an example of the image. This is a typical case of an example of a satellite image.

(Refer Slide Time: 19:10)

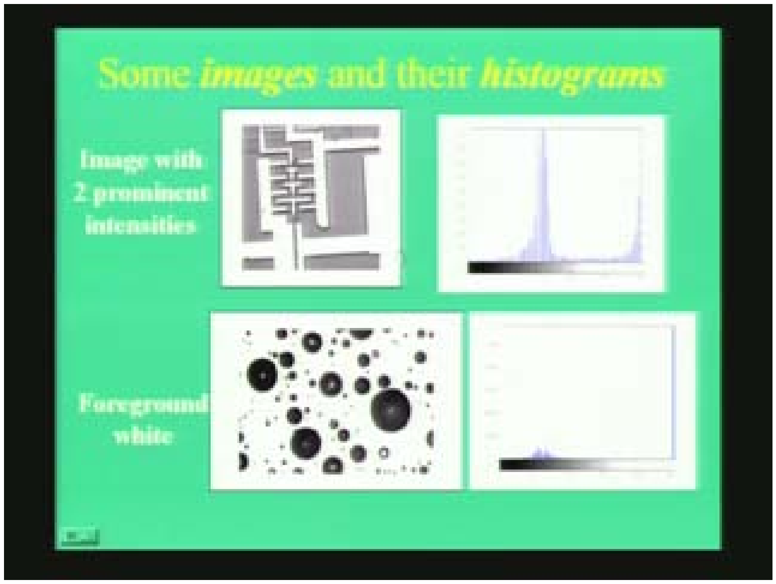


The histogram of the image here is given below see. If you look at this plot on the bottom of the x axis you have the gray level values rk. Of course I must admit here that the value is not normalized it runs from 0 to about 256 and there is a horizontal band of intensity as you can see here which is running from very dark to bright values.

So the darker values are from 0 to about 100 and brighter values are from about 150 to 256 the maximum values. See if you divide this range by the maximum value 256 which is l and the number of gray level values then of course you will get normalized range rk 0 to 1 and the vertical coordinates display the nk number of pixels with gray level value rk.

So it is basically a plot of nk versus rk unnormalized. In this case the probability to normalize both the axis you have to divide the vertical values on the vertical axis by n and on the horizontal axis by a value l otherwise this is a histogram value of nk versus rk unnormalize. That is the histogram example which says that at what is the count or number of pixels which exist in these image with a certain gray level value.

(Refer Slide Time: 21:18)



So you can see that there are almost negligible amount of pixels with very low very dark pixels and there are a few brighter ones but most of them is somewhere in the middle. The images are of moderate quality. Almost we can infer by looking an histogram that the image quality is not bad because the entire dynamic range of the pixel values are covered and we will see that these image in general does not require enhancement. But just to understand what the histogram is that is the example. So the plot of pr of rk with respect to rk is called the histogram of the image that is the definition of the histogram. So in this case of course I must inform you that you are plotting nk versus rk unnormalized range but you can easily transform this to normalized value. So that is the histogram.

Let us look at some more examples of images and their histograms. as you can see here for this image this is a image borrowed from and of course I must say here that these examples have been generated using the Matlab simulation toolbox, image processing toolbox and you can see this is a image of a printed circuit board layout it has two different gray levels the bright and the dark one and histogram also says the same.

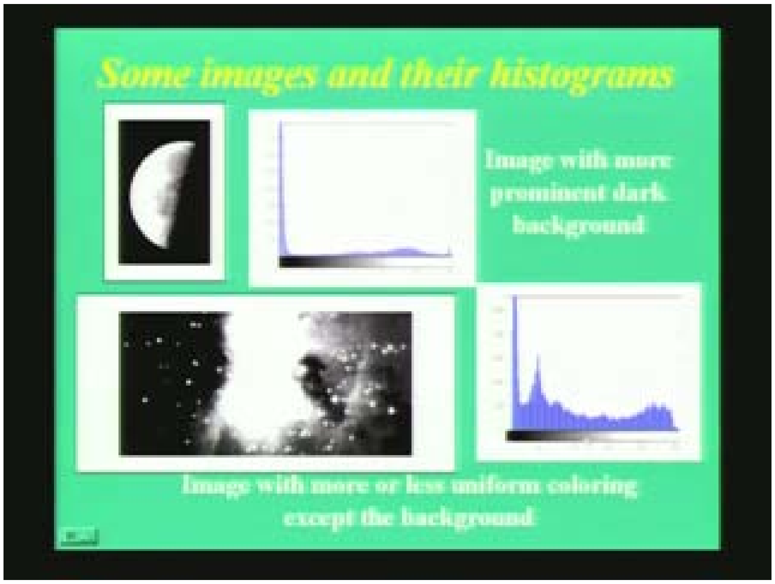
So you have pixels which are very bright here that is the probability here. of course the actual number of occurrences are provided here and there are of course some pixels which are on the darker side somewhere here this is the darker range and this is the brighter range.

Look at this image as for example which is a foreground white image the previous one image are two prominent intensities. The next one has a foreground white and a considerable amount of dark pixels which accumulate here in terms of its probability or occurrence. And you can see the last vertical line here which tells that the brightest pixel has a huge amount of pixels which are bright all over this image.

Some more examples, this is an image with more prominent dark background. If you remember the previous image was a case of white background, this is a dark background so this is the background which is dark and the histogram also shows it required a large number of pixels with darker intensity and of course there are a few in the brighter region as well but quite a few, these pixels corresponding to this particular region of the image.

You can see here this is an image with more or less uniform coloring except the background. So this is a very large background which occupies a large portion of the screen which is occurring here in terms of the histogram. And of course there are quite a few pixels on the brighter side which are occurring here in the image and their histogram shows here.

(Refer Slide Time: 22:31)



We had seen about four different examples of images and their histogram and I must also tell you that what does the histogram give you? It gives you a probability plot that it just tells you the number of occurrences of a gray level value within an image.

It does not tell you anything else as what information it contains, is it the image of a face, is it the image of a printed circuit board or is it an image of an Ariel view of a city or the sky, no that information is not presented in the histogram. You should keep in mind that the histogram gives or reflects the global appearance not any structure and information or the content information, the global appearance in terms of its quality of the image.

It might tell you that well the image appears to be on the darker side or it could be on the bright brighter side you can look at the histogram and tell that. You can look into the histogram of an image without looking into the image and say well it seems that it is a very dark image.

You could say that image is probably very bright. Or you could also say that the dynamic range of the pixels is quite wide that the image probably is of good contrast. It may happen when the perfect histogram should be a flat end histogram which we are trying to do by the process of histogram equalization which we will see now.

So we look at different applications of histogram now or the use of histogram. As you say here the histogram are simple to calculate we have seen the formula of the histogram just now before we looked in to the examples.

It gives information about the kind of image or the global appearance and its properties. Remember this global appearance is a very qualitative term. It does not tell you about the structural contents of the image whether the image is of a classroom or an outdoor scene or a photograph of a flower or a garden, no. There is absolutely no information about that in the histogram of an image. It just gives you the global appearance in terms of the lighting distribution or gray level distribution probably the distribution and that is all you have in the histogram.

(Refer Slide Time: 25:51)

But you can work out magic using that and people have used histogram for many different applications which we may not have time but we will see how to enrich the contrast of an image or by using the histograms.

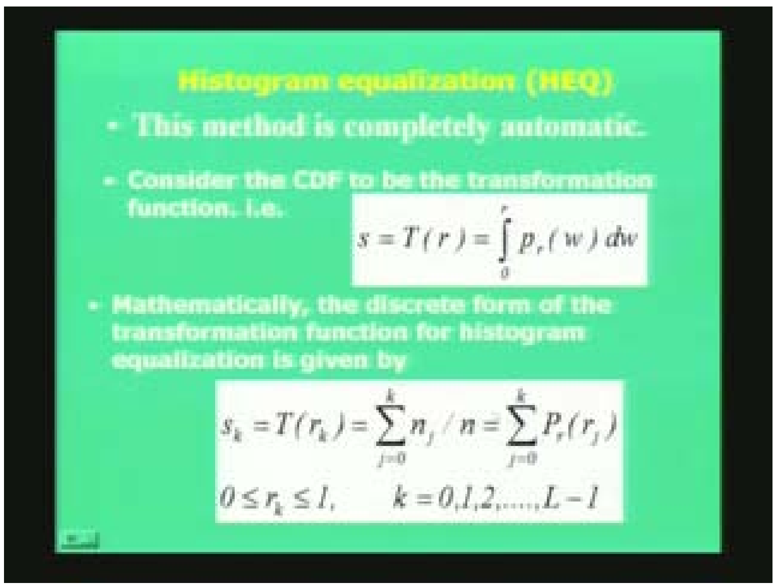
So the histograms are simple to calculate as we see here it gives information about the kind of the image and its properties. That means basically global appearance is used for image enhancement which we will see here next in terms of histogram equalization. It is also used for image compression not to a great extent but yes it can be.

The other big application is; it is used for image segmentation and it can be used for many real time image processing applications as well.

We shall now look at histogram equalization which is the main point of discussion now. And here the goal is to obtain a uniform histogram for the output image. What do you mean by uniform histogram and how do you obtain it by using histogram equalization.

What is histogram equalization? The method is completely automatic. Even computing the histogram and performing histogram equalization both are completely automatic. And to perform histogram equalization you must consider what is called the CDF or Cumulative Density Function or distribution function CDF of the transformation function in this form s equal to t r is what we need to obtain in terms of contrast stretching. And if you are using the histogram equalization to do contrast stretching this s equal to t r function must be evaluated using these integral form 0 to r pr of r we know that is the histogram of the probability, w in this case is just the dummy variable for integral and this is just the histogram. That is the formula for histogram equalization.

(Refer Slide Time: 27:04)



I am going over a few mathematical concepts here keeping the equations as simple within the time limits available to us. Mathematically the discrete form of this analog or integral equation of the transformation function for histogram equalization is given as this.

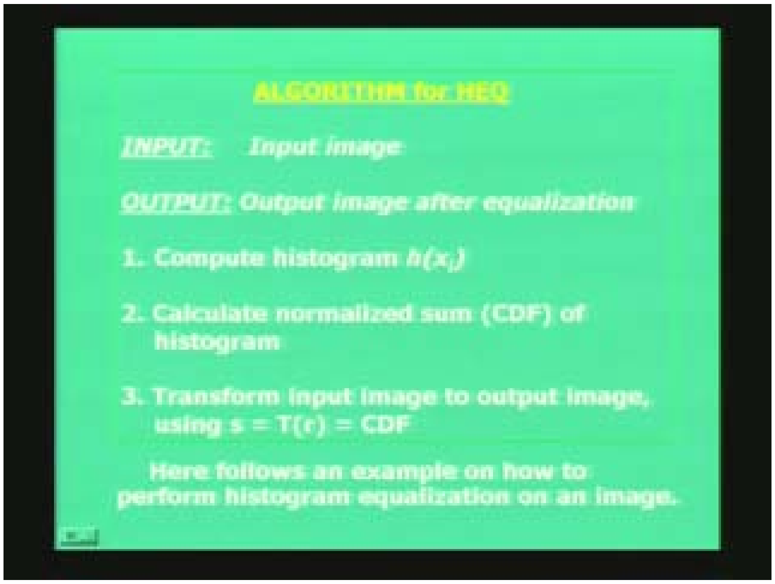
Remember, the integral is a sigma in the discretized world so that is what this is here as you can see and this is nothing but the histogram nj by n. So sk t of rk remember that is the discrete form k is integer indices running from 0 to l minus 1, l for l different gray level values, rk normalized range 0 to 1 and sk equals t of rk is nothing but this summation which is nothing but this and this sigma pr of rj sigma 0 to j running from 0 to k pr of rj is nothing but this particular integral. This integral in analog form is discretized here. The discretized version of this integral and this if you substitute from the formula of the probability distribution or histogram gives you this.

So remember this is what you have to compute summation of k plus 1 terms for sk equal to t of rk and that you have to compute for nj by n. That is the formula for histogram equalization and you have to basically sum the terms.

We will look at an example, we will look at the algorithm steps in fact for histogram equalization first which will help you to take a overall look at it using this formula and then we will look at examples, one example in a classroom exercise to see how the calculations could be performed and then we will see examples on images.

So the algorithm for histogram equalization or HEQ in short is the input image is given, output is the input image after equalization.

(Refer Slide Time: 29:35)



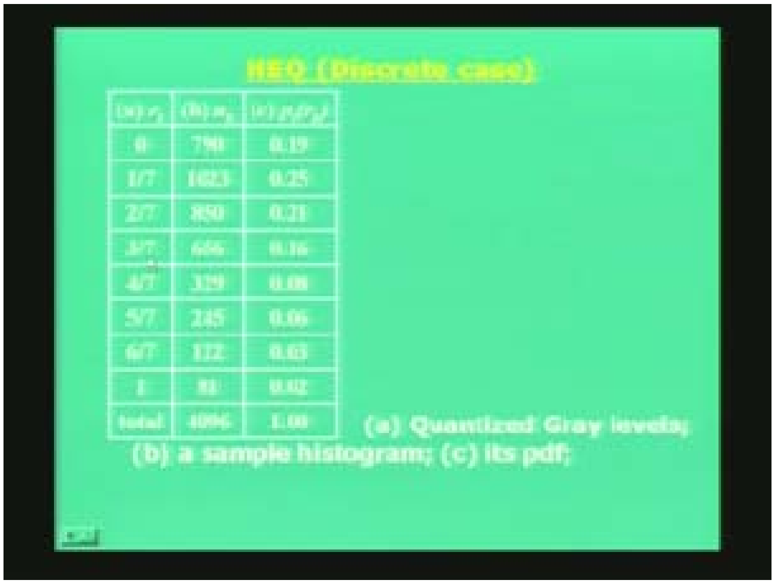
Of course the first task is to compute the histogram h of xi of the input image itself. For the input image compute the histogram, calculate, normalized sum of histogram that is another way of calling it or it is also called the CDF normalized sum of the histogram or CDF given by the formula earlier and that computed CDF is nothing but your transformation function s equal to t r. So transform input image to output image using this transformation function s equal to t r where this s equal to t r is nothing but your computed CDF. And what was the formula for CDF you remember? This was the formula for computing the cumulative density function or distribution function for the input image so that is what you have for your algorithm.

So whenever your transformation function s equal to t r is equal to the CDF of the image or what is called as the normalized sum of the histogram then that gives you the histogram equalization that is as simple as that. It is easy to implement the formulas are very easy as you can see here and we will start to look at an example on how to perform histogram equalization on an image.

So let us look at an example in the discrete case. I encourage all people to work out this example in your paper as we go along, just check the values. As you see here rk there are 8 gray level values normalize 0 to 1 so 0 1 by 7 2 by 7 and so on.

So there are 8 values so k will run from 0 to 7, 8 discretized gray levels running from 0 to 1 and the gray level occurrences of value 0 or 790, 1 by 7023 and so on. And if you sum up all of these you will get a value 4096.

(Refer Slide Time: 30:07)



I leave it as an exercise for you to check up that this 4096 which is the total number of gray level values if the images is a square array can you guess what is going to be the size of the resolution of the image? How many rows by how many columns the image will have? The total number of pixels of gray level is 4096. It is easy for you to guess it is 64, so 64 into 64 gives you 4096.

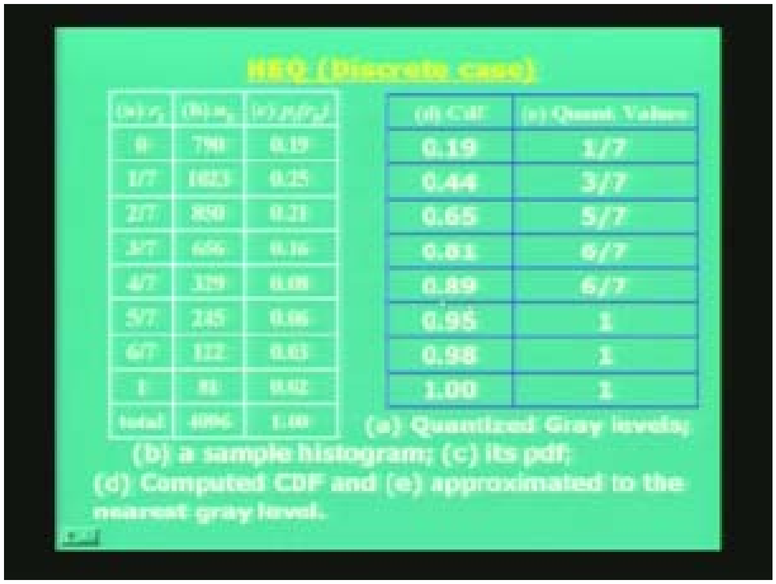
So after you compute the histogram you should ensure that the summation of all these values is this particular value total number of pixels in the image. And this is your normalized p histogram that is pr of rk.

How to get this value? Divide 790 by n, nk by n this is your n 4096 total number of pixels in the image. So nk by n gives you this value.

So divide each of these first eight values by this n and this is what you get of pr of rk. And you must again verify that as the sum of these values gave you 4096 the sum of these value should give you 1 if it is not so, if all these fractional values is do not add up to 1 what is the sum of these fractional values in terms of the plot of pr of rk?

If you look at the plot of pr of rk what is this rk? We will see this plot later on, it is the area under that curve, that area under the curve should be equal to unique because the sum of all probabilities. Sum of all probabilities is 1 so that is what you have here pr of rk sum of these values is equal to 1.

(Refer Slide Time: 32:51)



Now, a is the first column quantized gray levels, b the sample histogram which you have taken as an exercise and its pdf or the histogram is given in the column c the third column.

So let us go ahead and start the computation of the CDF for this particular example and try to equalize this image. Of course we are not bothered about the image here itself it could be any image for which the histogram is computed here. And we will look at the case of how to equalize the histogram that is our main exercise

So look at CDF, how to compute the CDF? As you can see I was telling you earlier in the expression that is what you have to compute the cumulative density function it is sum of the value so 0.9 here 0.9 plus 0.25 is 0.44, 0.44 plus 0.21 and 0.65 so 0.65 plus 0.16 is 0.8 and so on. At any position or row of the CDF this particular value will be the sum of all the previous pr of rk values from the beginning rk equal to 0 to that particular value k. And you must also reach a value 1 at the end.

Actually if you do not reach a value 1 here you will not reach here. Or as you keep on adding this 0.25 to 0.19 to obtain the next value 0.21 to 0.44 to get the next value and so on you should also reach the value 1 at the end. If you do not reach a value 1.0 or 1.0 there you would have definitely made a mistake in the calculation somewhere in computing this histogram.

So that is the computed CDF value and what are the quantized values? The quantized values are given here. How to compute these quantized values? Well, you can see we are working in the discretized domain and in the discretized domain we do not have all the values from 0 to 1 these are the only possible values. So we look at the fraction of one of these fractions 1 by 7 and 2 by 7 and so on which is nearest to this decimal value 0.19 which is the closest and that is what we assign. That is the approximation to the nearest gray level value is in the last column e where we have approximated this decimal value to this quantized value. So 0.19 the nearest quantized value or fraction is 1 by 7.

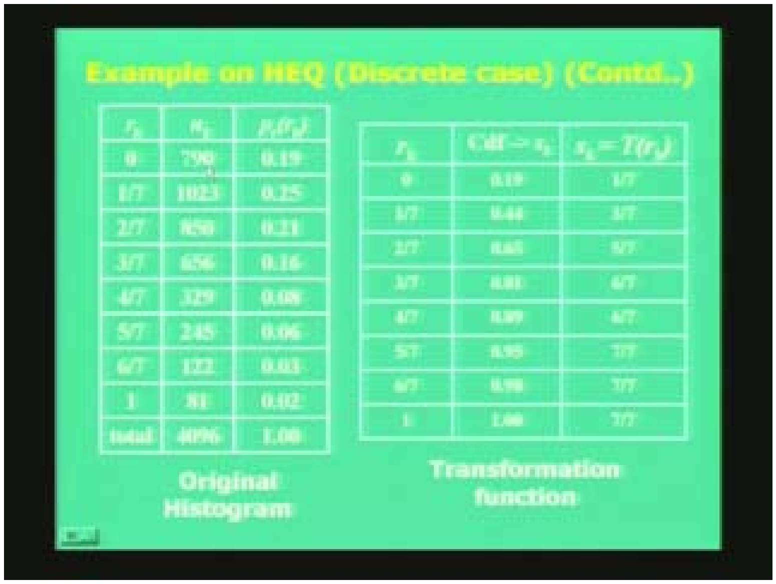
The nearest quantized value for 0.44 a decimal value here is 3 by 7, 5 by 7 is the nearest fraction for 0.65 and so on and all these will actually round of to the nearest quantized gray level value of 1.

You have not computed the histogram equalization but this is how you computed the CDF at least. CDF in analog sense and in the discretized domain also we must actually see what is the quantized level values here because CDF could appear as a continuous function after these fractional values and none of these could actually be equal to a quantized gray level value that is possible but you have to do this assignment in terms of quantized steps.

These are the different quantized steps possible of values of rk, the quantized gray level values as given here in column a and you have to find out any value of CDF here the decimal value which is the closest fraction out of these eight possible quantized gray level values and put that value here. So that is what is done you can check it for yourself that these will be the values obtained from CDF.

Let us go ahead and do the equalization. So where were we in the last slide? This was the original histogram. If you remember, this we had explained in the previous slide, gray level values, gray level occurrences and the probabilities which is the histogram the original histogram here and this was that for an rk now we have combined the previous two tables in the previous slide and for a value of rk cdf points to sk which is this and the nearest quantized level value is given here. So this is the transformation function from rk to sk which will give you the histogram equalization.

(Refer Slide Time: 36:12)

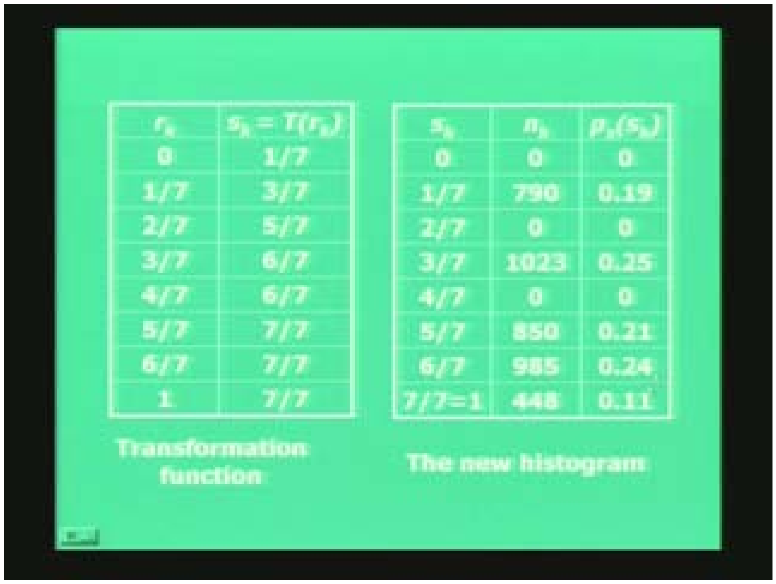


So it basically means that all gray level values with 0 in the original image should be transformed to 1 by 7, all gray level values 1 of 1 by 7.

How many of these are 1 by 7? It is 1023 of them, 790 pixels with gray level values 0 will be transformed to 1 by 7, 1023 pixels of gray level value 1 by 7 will be transformed to 3 by 7, 850 pixels with gray level value 2 by 7 will be transformed to 5 by 7, 656 pixels with gray level value 3 by 7 will be transformed to 6 by 7, 329 pixels with gray level value 4 by 7 will be transformed to 6 by 7 and so on and that is how it is done.

That is the transformation function and that is what we used to equalize the image. Finally this is the transformation function rk to s of k and the new histogram is what we have here because sk you can see that the transformation function may not have certain gray level values. There are no pixels with gray level value 2 by 7 or 4 by 7 as computed earlier in this table.

(Refer Slide Time: 37:00)

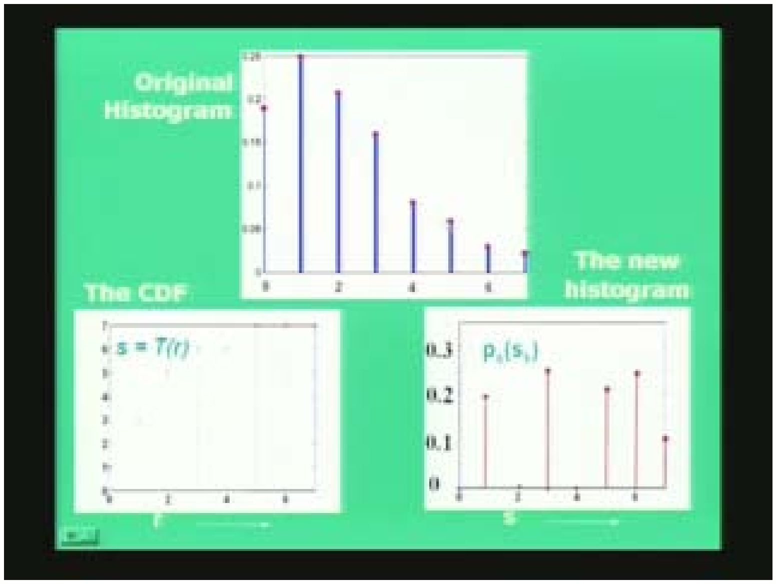


And 790 pixels had a gray level value 0 and those have been assigned 1 by 7 and that is the probability. You should see that the some of these values should also add up to 1. You can do that calculation right now, 0.19, 0.25, 0.25, 0.21, 0.24, 0.11.

Some of these five in fact if others were also non 0 you have to add all of them. Summation of these probabilities as we have done for original histogram for the original image for the transformation transformed histogram equalized image also the summation of these values should be equal to 1 and this is the transformed image. So this is how you do the histogram equalization.

Let us look at the plots. This was the original histogram. If you look at the plot well we had a 0.19, 0.25 and the decaying value. You can look at the values here given in the previous slide let us go back. This was the original histogram. These were the values and this is what I am showing you in the plot in the next slide here.

(Refer Slide Time: 38:07)



Let us see this is the plot. What would you infer from this histogram which we are going to equalize or we have already equalized in the table? We will show the plot now. So this is the pr of rk versus rk. First of all it is not equalized there is a scope of equalization here because the histogram does not appear to be flat. There are more number of pixels with darker gray level values.

The image might appear darker so we are trying to enhance the image by adding more and more gray level values. But we cannot do it in a random fashion just simply transform some of the darker gray level values to bright will not solve any purpose value. So we are doing it in automatic fashion using probability distributions of the histogram or the histogram and the CDF equal to the transformation function s equal to t of rk when it is equal to CDF that is your histogram equalization algorithm.

Let us look at the cumulative density function of this image. Let us see how it looks like? That is the plot. So any typical CDF of an image will look like this. Of course the nature by which the curve the points of the curve go up will vary from one image to another. But it will monotonically keep increasing all the time. That is the property which is used are exploited to perform histogram equalization so the values go up and this histogram is the constructed form and we know how to do that. The table of values is also given in couple of slides back and that is your CDF of the image of the transformation function used for histogram equalization. And when you do that this is the new histogram.

As you can see here some of the values have disappeared and there no gray level values, no pixels with the gray level values 0, 2 and 4 but you can see this is much flatter compared to the histogram which has a very exponential decaying nature here. Large number of pixels is dark and hardly any pixels are bright.

Now you have quiet an average distribution with where definitely there are pixels in the darker range and some in the brighter range so that is the histogram equalization example.

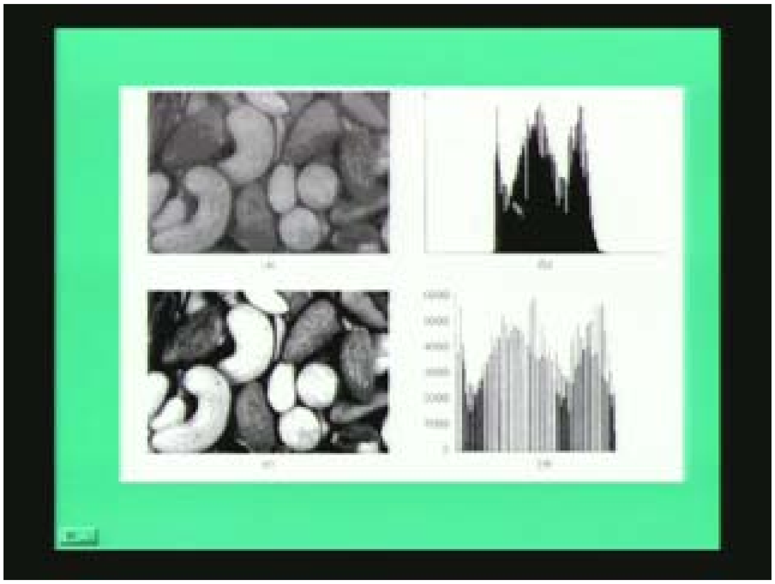
Well I am sure there are lots of questions which come into your mind. The first question is do you gain anything and really do you think that this histogram is equalized or it has flattened out? It has but it does not appear so neat and nice. Well that is due to the discretized nature of the data. It is due to the finite number of gray level values we have, finite number of pixels we have in the image when we move out from analog domain the equation to the discretized domain and we are doing a lot of rounding operations of some pixels.

Some pixels might disappear and actually some pixels may be reintroduced. It may happen in the discretized domain that some pixels may be reintroduced back which may be absent in the original histogram. So that is possible in the histogram equalization and that is why but the plot of the equalized histogram you see is not truly flat and some pixels are missing due to two principal reasons that we have considered a very small image with a finite number of gray levels and pixels.

The larger the resolution of the image you have larger number of gray levels you have practically if possible, that is not possible. Hypothetically if both are at infinity or as large as possible value you can imagine then only you will have an absolutely flattened histogram with all gray levels possible. That is the reason why you see that the new histogram here appears to be quite flat but it is not flat. Definitely it’s much more flat than the original histogram and of course some pixels are increasing.

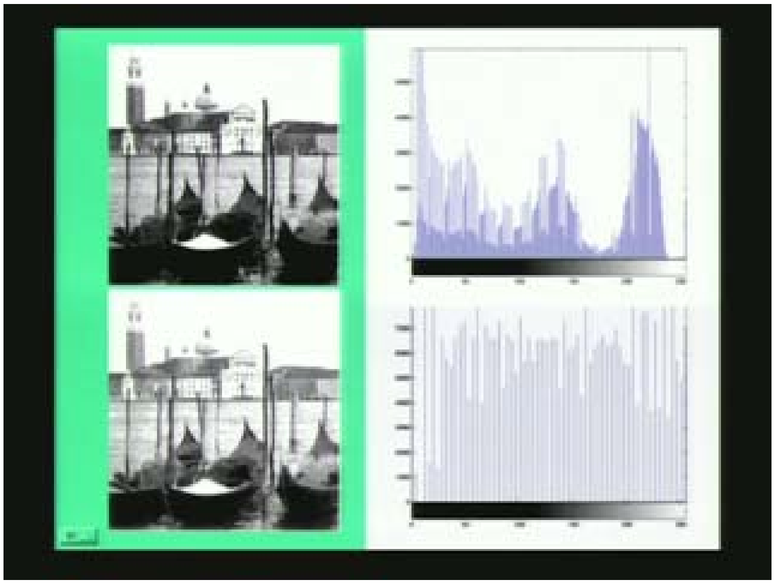
Let us look at some examples. This is a very cuter example, as you see this image is not that rich in contrast this is the original histogram of the image or histogram of the original image and there are no pixels on the darker and brighter side.

(Refer Slide Time: 41:41)



You first equalize in this range and then this is the output of a transformation which you see that this is much richer in contrast then this particular one.

(Refer Slide Time: 42:29)



Let us look at another example, this is an image of the city of Venice and this is a histogram. Well, it is not flattened but you do have dynamic range of pixels from dark to very bright ones except the bright ones are absent in this but it is not flattened. So when you try to flatten this the histogram will appear something like that. As you can imagine here that it is flat but not absolutely flat but of course one or two pixels may be missing as well and you may try to flatten this and the output image will appear like something.

If you observe very carefully there is a little bit of enhancement, there are some very dark regions which have become a little bit brighter here.

Some details which are very dark around these regions or even the water body as you can see here the Venice palace is on the back side it is a city submerged in water mostly and the foreground here are very dark regions but are little bit brighter including the water body. So that is the change which one can make about this image which is quite good in quality and not that bad. So histogram equalization tries to do its best but it is not always good.

We will see an example here, that is a very good quality image because it is used for all examples starting from image compression, segmentation, enhancement.

(Refer Slide Time: 43:30)

You do not need to enhance this image but if you try to enhance this image you can almost imagine what is going to happen. This is the example of the histogram it is not flat but it is quite broad spectrum histogram in the sense that there is a lot of dynamic range of pixels present here and you flatten it out and this is the output which you have.

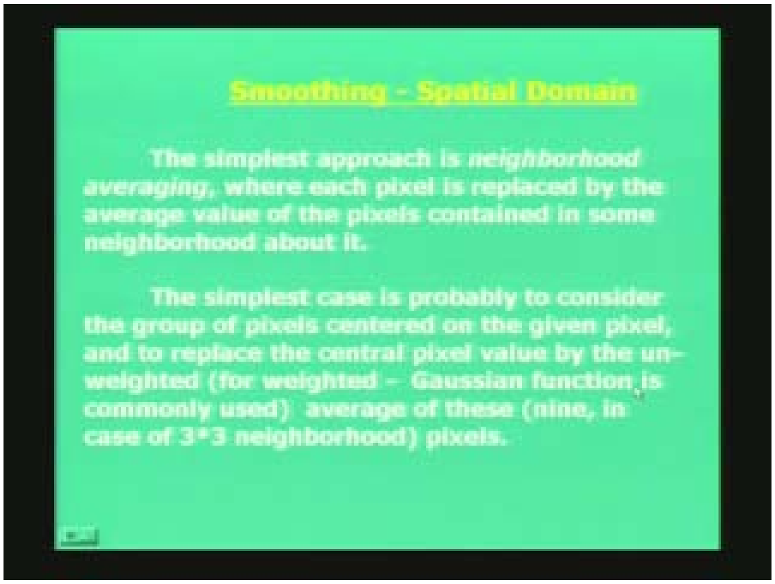
There is hardly any difference. In fact it has worsened the quality somewhat. If you carefully observe that this image is somewhat saturated. So sometimes do not try to perform histogram equalization on an image which is quite rich in quality by itself.

There is another example where the histogram equalization should be avoided. So that is the case of examples we have seen about histogram equalization and we will move to the last section of this lecture where we talk about noise removal or smoothing using smoothing operations in an image.

Let us look at smoothing in special domain. The simplest approach is what is called the neighborhood averaging where each pixel is replaced by the average value of pixels contained in some neighborhood about it.

And the simplest case is probably to consider the group of pixels centered on the given pixel and to replace the central pixel value by the unweighted average of these. When you talk of weighted we can use a Gaussian mass or a Gaussian function but it is unweighted average of 9 pixels in a 3 into 3 neighborhood.

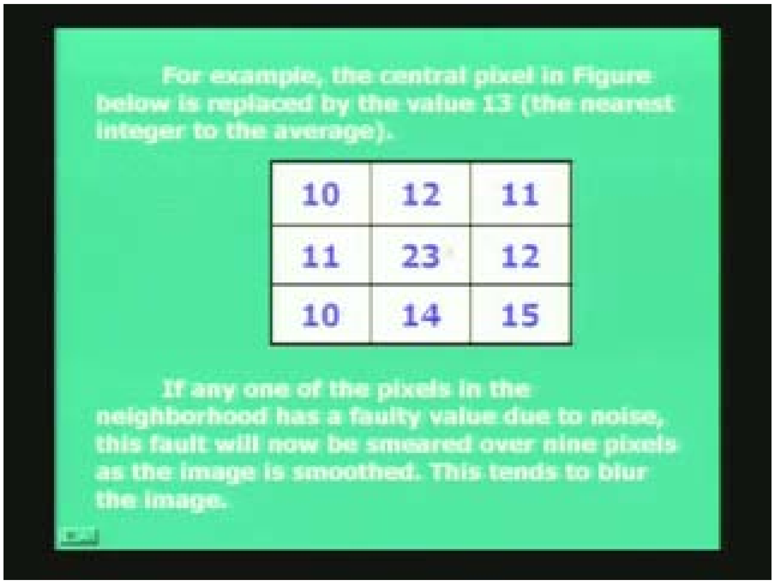
(Refer Slide Time: 44:25)



Let us look an example of what this means. For example the central pixel in this figure given below is the value 13 that is the nearest integer to the average. If you take all these 9 pixels in a 3 into 3 template or window or mask as it is called average of these nine values 3 into 3 equals 9 values here will give you 13.

As you can see this image it appears that these values should have been 13 or someone need to 10 and 11 or lesser than 15 or 14 but this is probably due to a effect of noise which has caused this pixel to shoot up well above its values around it.

(Refer Slide Time: 45:20)



So you can average it and replace it by its value and in this case it works. So if any one of the pixels in the neighborhood has a large or a faulty value due to noise this fault will now is smeared over all the nine pixels as the image is smoothed this tends to blur the image. That means this value will not only effect this pixel but when you do averaging around the image of all these other pixels the neighborhood pixels also will be affected.

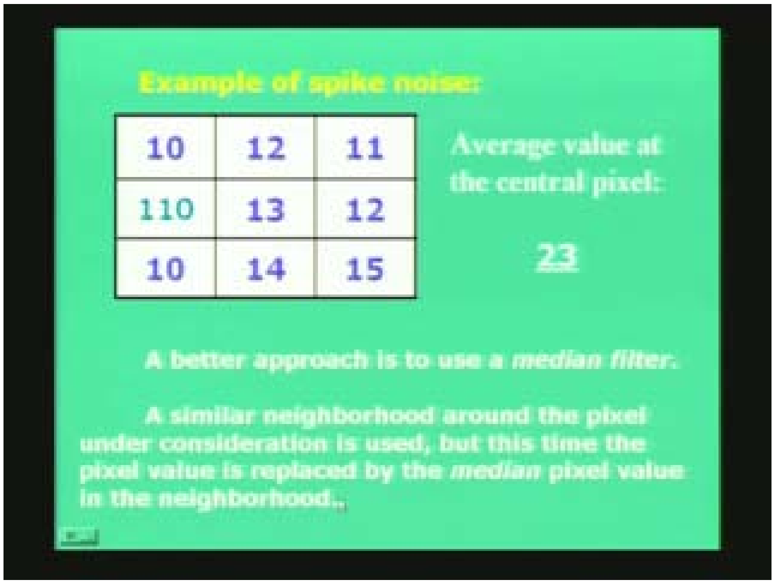
Of course this is the case of 3 by 3 you can average using 5 into 5 or 7 into 7 window, the effect of blurring will be still larger.

Let us look at the different type of a noise which is called a spike noise here. Let us say this central pixel is replaced by 13. It was that noisy in the previous case as 23 so that I replaced with the original value. And the pixel on the left which had a value 11 I put a large amount of what is called a spike node. That means this is so large when compared to its neighborhood that is definitely it is something like an impulse or a large spike which has come so it is called a spike noise.

If you see the average value of the central pixel it gets corrupted. it is the value 23 which you get as an average so it corrupts not only this but if you try to compute the average around here or here anywhere surrounding this spike noise it will effect the neighborhood also. So the averaging effect or the effect of averaging using special domain is not a very good method to remove noise. It works to certain extent but it does not work beyond a certain point especially it is very worse in the case of spike noises.

So a better approach is to use a medium filter where a similar neighborhood around the pixel under consideration is used.

(Refer Slide Time: 46:51)



But this time the pixel value is replaced by the median pixel value in the neighborhood. We are proposing a technique of median filtering which is also very popularly used.

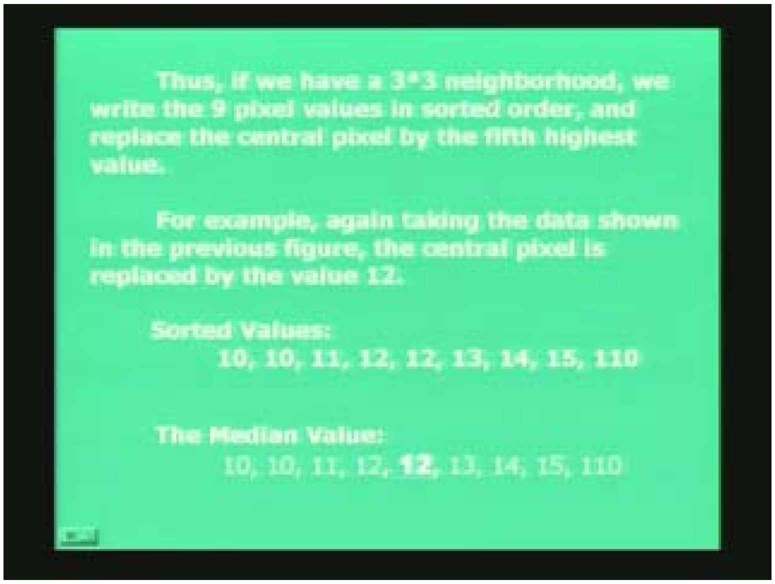
So instead of spatial averaging where we took a set of pixels around the neighborhood and compute it with a simple average and we know the problem using averaging now. It is a very common simple technique averaging can be used. We replace it by a concept called median filter. So you will understand this concept of median filtering now.

Thus if we have a 3 into 3 neighborhood we write the 9 pixel values in sorted order. So what is the method of median filtering? I read again; we are discussing the concept of median filtering where if we have a 3 into 3 neighborhood we write the 9 pixels values in sorted order and replace the central pixel by the fifth highest value. I am taking a 3 into 3 neighborhood and that is why I have 9 values but nobody stops you from taking a large neighborhood.

You can have 5 into 5 then you have 25 values which you need to either average in case of averaging or when you doing many filtering to sort those 25 values and the median in that case for 25 values will be the thirteenth value.

In the case of 3 into 3 neighborhood when you have 9 values the median is the fifth value we have seen that. In the case of a 5 into 5 neighborhood when you have twenty five values the median value is the thirteenth one. If you choose a larger window 7 into 7 let us say you can start guessing why I am talking about odd numbers not even 4 into 4 equal to 6 into 6 that is easy to guess. I leave it for you to think, 7 by 7 equals 49 gray level values what is the median 25th. But for simplicity here in order to handle the values in the screen we are talking of simply 9 pixel values and we replace the central pixel by the fifth value. So, for example again taking the data shown in the previous figure the central pixel value is replaced by 12. If you have sorted value like this then what is the median value? It is the first second third fourth and fifth this is the median value. So you just do not worry about it, the median value is 12 here which is highlighted and underlined here.

(Refer Slide Time: 48:59)



So as you see here from the previous template 3 into 3 neighborhood gives you 9 pixel values then you need to sort them as when you sort them as you see at the spike noise.

If it is a lowest spike noise so low value or even a large value here the spike noise will be shifted to the end of that sorted order and it will be either at the beginning or the end depending upon whether you have a small or large value take the median you will be safe and that is going to be close to the actual true value of the set of pixels so that is what is median filtering, take the neighborhood sort all of them select the median value that is simple. And you can do it all over the image and it works reasonably and fairly well definitely much better than the neighborhood average. So the approach has two advantages as we see here in the next slide. Occasional spurious high or low values are not averaged in they are ignored that means spike noise is thrown out.

This spike noise causes occasional spurious high or low values of gray levels and those are not averaged in. You know the spurious spike noise with a very large or small value will affect the average value. That is what it was doing and also it will start spreading and cause an effect which is called blurring in the image. Larger the mask from 3 by 3, 5 by 5, 7 by 7 or even higher 11 into 11 or even 23 by 23 or even larger if you visualize the larger window size you take to operate in able to averaging the more effect of blurring.
But the median filtering does not have that median filter does not have you can take even larger size and if the larger size window captures spike noise pixels having two or three different spike noise at two or three different places since you are sorting them those values will be the extreme end at the beginning or the end of the array of the sorted sequence and you pick up the median value. That is what I say as the occasional spurious high or low values are not averaged in and they are ignored. And the other important

advantage is the sharpness of edge is preserved in this and to see an example let us consider the pixel data as shown below which will say or illustrate what do you mean by sharpness of an edge.

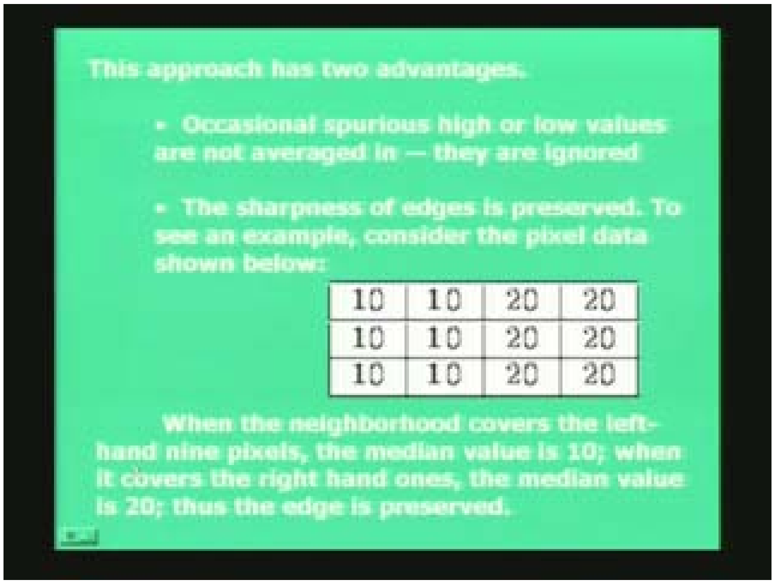
This is an edge where you see that the pixel value to the left of an edge I am drawing the edge in the vertical direction here or certain constant value is here to the right and they are different and I want to preserve.

Now, if you can imagine here if I do a 3 by 3 averaging around these pixels here this 10 value will go up and the 20 value will go down bringing the sharpness of the edge which will disappear. The sharpness of the edge will not exist any more and to preserve that what the median filtering will do? When the neighborhood covers the left hand 9 pixels the median value is 10 anywhere here and when you cover the right hand ones the median value is 20 and hence the edge is preserved.

You can take any central pixel take a 3 into 3 neighborhood suppose this is the pixel you are considering the value is 20. And you take a 3 into 3 sort these values take the median you will have a value 20.

Let us take this pixel we take a 3 into 3 set of pixels here replace it by the median one you will have a value 10 here so that is what we are reading here. The neighborhood covers the left hand 9 pixels which is these three we have the median value 10. When it covers the right hand ones that means these set of 9 pixels 3 into 3 are covered the median value is 20 which will be replaced here and hence the edge is preserved.

(Refer Slide Time: 51:43)



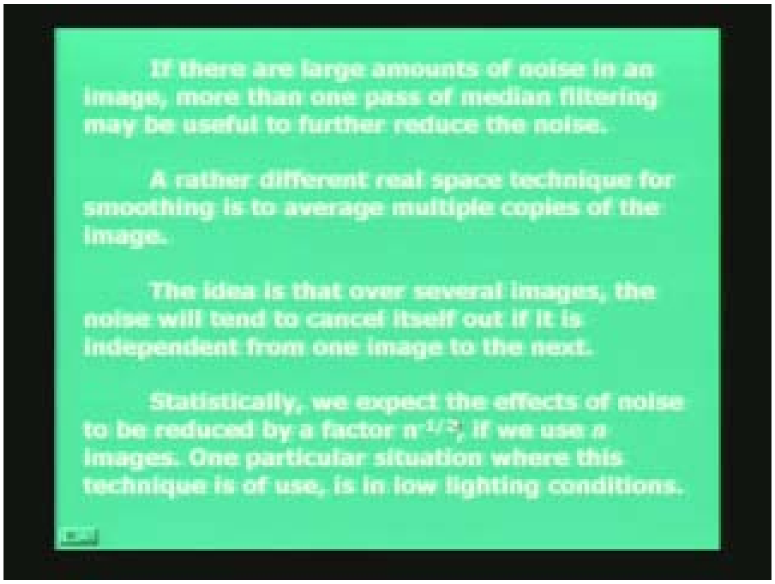
So it is two important properties advantages of median filtering, it preserves the edges and it removes the spuriousness.

There is some information about noise. If there are large amount of noise in an image more than one pass of median filtering may be useful to further reduce the noise you can use such a concept. And a rather different real space technique we are not talking of any spectral domain techniques here for smoothing is to average multiple copies of the image.

Where do you have multiple copies? Well, if we are talking of a video where the inter frame differences are not much you can actually or there is hardly any movement or any negligible movement or in fact with possible no movement static face video taken and you have lots of samples few tens or few hundred of them and even if there is noise you can actually average it out across time frame.
You get a better image frame which is free of noise. So that is what is talked about here, average or multiple copies can be done.

The idea is that over several images the noise will tend to cancel itself out if it is independent from one image to the next. And statistically we expect the effects of noise to be reduced by a factor of 1 by root over n if we use n images for particular situation where this technique is of use is in low lighting conditions.

(Refer Slide Time: 53:33)



So let us look at an example of median filtering and compare that with the averaging within the remaining time we have.

We have seen two types of techniques now of removing noise. One is averaging which we know is not that powerful especially with respect to spike noise, it degrades the sharpness of an edge, it is unable to handle spike noises and we have seen the median filtering which is powerful. So let us look at the original image Lena again and I have corrupted this image by adding noise. You can see that the image is quite noisy compared to the image which is very clean here, absolutely this is the original image I have just corrupted, you can do that using any simulation toolbox. I have used Matlab simulated to do that which is very simple and I apply the median filter image. You can see that the median filter image is quite close to the original image in fact it is much better with respect to the noise image.

(Refer Slide Time: 54:28)

Let us look at another example of an original image. This is the noise image, in fact this is the spike noise. The previous case was a salt and pepper noise, this is the spike noise here and you can see these spikes can be removed by median filtering. There is a little bit of defocusing or blurring effect but the median filter image is much better than the noise image and in fact it is quite close to the original image.

So these are the examples of methods by which you can filter an image. and of course you can I leave this an exercise for you. You can use any simulation toolbox or write a program yourself, read an image in bitmap form and then try to remove the noise by average filtering and then try median filtering and see which one is better.

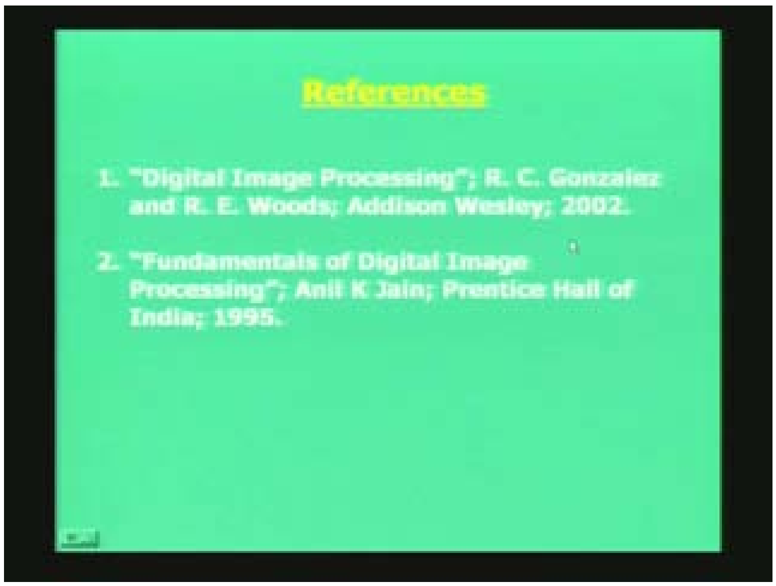
I can almost guarantee you that in most situations the median filtering will win. Of course there are many sophisticated techniques which are even better than median filtering. But we have no scope of that in this course.

We have to look for techniques based in courses like digit digital image processing which will discuss more of these aspects here but when you need to just clean of an image for use in computer graphics applications for image based rendering or when I want to map an image on a certain background it is better that you clean it if necessary if there is noise in the image.

This completes our discussion on image filtering or image enhancement. First we have discussed contrast stretching then by transforming the gray levels and then we discussed at length today what is histogram and histogram equalization based method and then of course noise filtering by smoothing and median filtering.

I would want you to look and note down the references which I used for this particular part of the lecture. And this is also applicable for jpeg compression and the recent book by Gonzalez and Woods has a chapter on image compression. I would encourage you to read that and these are the list of references about digital image processing.

(Refer Slide Time: 56:34)



That brings us to the end of the lecture series on computer graphics as well as image processing. So we today we wind up digital image processing, compression and enhancement and the two aspects which we discussed in the last two lectures.

In the last lecture it was compression, it was enhancement in this lecture and that brings us to the end of the entire series of lectures in computer graphics. I hope you have enjoyed all the lectures and understood them. But I encourage you to go back, read books the references which are given for both computer graphics and the last two lectures on image processing and then try to write some programs may be using OpenGL which we studied as well the couple of lectures on OpenGL and try programming them.

Of course you can use some simulation toolboxes for graphics using VTKS Virtual Reality Toolkits you can use. But I will ask you to do programming as computer science engineers and scientists and experience the real world of computer graphics and also virtual reality, thank you very much.