Computer Vision and Image Processing- Fundamentals and Applications
Professor: DR. M. K. Bhuyan
Department of Electronics & Electrical Engineering
Indian Institute of Technology, Guwahati
Lecture-19
Color Image Processing

Welcome to NPTEL MOOCs course on Computer Vision and Image Processing: Fundamentals and Application. In my last class I discussed the concept of color image processing. So, there are two techniques, one is that marginal processing and another one is the vector processing. In case of the marginal processing, the RGB components we can process separately, the R component I can process separately, G component I can process independently and B component I can process independently, that is called marginal processing.

In case of a vector processing I can consider as a vector pixel, because in a vector pixel I have three components, R component, G component and B component. So, all these components, the three components, I can consider together for processing. So, one is marginal processing, another one is the vector processing. In my last class also I discussed the concept of the full color image processing and one is the pseudo color processing.

In full color processing, I can process a color image just like a grayscale image, I can do image enhancement, I can remove noise of the color images. In case of a pseudo color processing that means the false color so, I can convert that grayscale image into the color image. So, that is the concept of full color processing and a pseudo color processing.

After this I discussed about some color models. So, one color model is X, Y, Z color model. After this I discussed the RGB color model, the CMY color model and CMYK color model. Now, based on this X, Y, Z color model, I have defined the chromaticity diagram. So, all the colors I can represent in the chromaticity diagram. That is about the X, Y, Z color model.

Now, I can consider the color has mainly two components, one is the brightness and another one is the chromaticity. So, one is the luminance and another one is the chrominance that is the color. So, chromaticity means I have to consider two components, one is the hue, another one is the saturation. Hue means the color corresponding to a particular wavelength. And what is the saturation? Saturation means the purity of color, that concept I discussed in my last class.

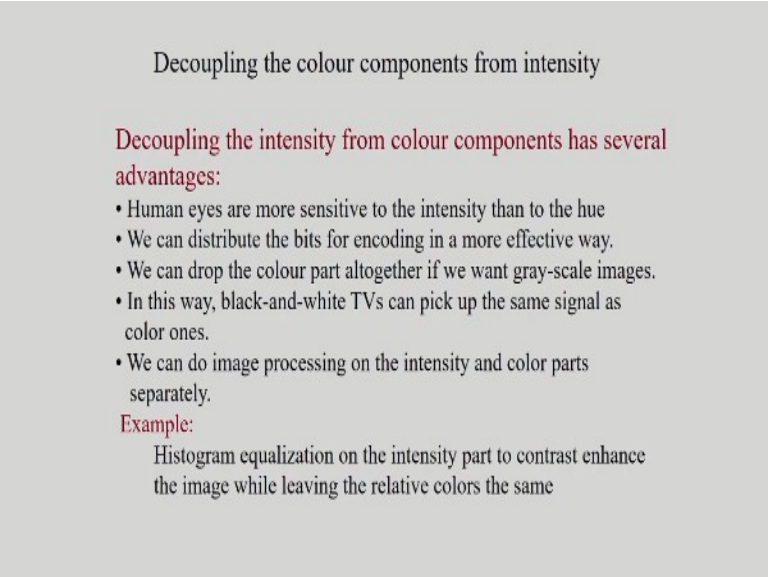
Today I will discuss some other color models, and the main concept is mainly the decoupling of color information from the intensity information, that is the intensity means the brightness information. So, I can do the separation of the color component from the intensity component and there are many advantages of decoupling.

So, one important application is in color image processing and that color image compression. Human eye is more sensitive to intensity variation as compared to color variation. So, that is why if I want to consider the color image compression, then in this case I have to allocate more number of bits for the intensity component as compared to the color component. And if I consider suppose the black and white TV transmission, then in this case only I have to consider the intensity component, I need not consider the color component.

Also, suppose if I want to do the color image processing, then in this case what I have to consider, only the intensity component I have to consider and maybe the color component I can process separately. So, for example, suppose if I want to improve the visual quality of an image, that is the image enhancement. So what I can do, I can process the intensity component only without affecting the color components.

So, these are the advantages of decoupling the intensity information from the color information in the color image. So, first I will discuss about this concept, the decoupling of intensity information or decoupling of color information from the intensity information. After this I will discuss some color models like HSI color models, YIQ color model, YCbCr color model. So, these concepts I am going to discuss today.

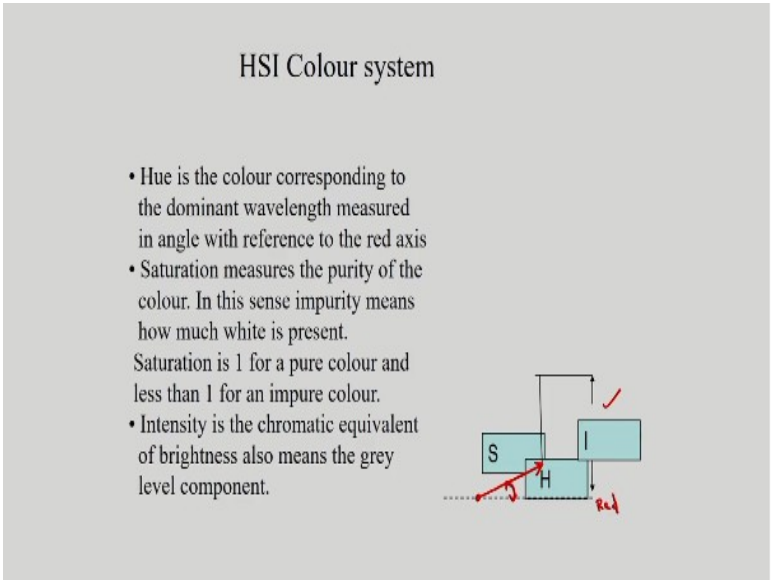
(Refer Slide Time: 5:06)



So, first one is the decoupling the color components from intensity. So, already I have explained this concept. So I can separate the intensity from the color components. And one important thing is, the human eyes are more sensitive to the intensity than the color, color means hue. So, based on this, suppose one application is the color image compression. So for this what I can do, I can allocate a more number of bits for the intensity component as compared to the hue component.

And the second example already I have explained, that is for a black and white TV. For a black and white TV I can only consider the intensity component without considering the color components. And for color image processing what I can do, I can only do the processing for the intensity component and maybe sometimes I can do the processing for the intensity component and the color component separately. So, this is the advantage of decoupling the color components from intensity.

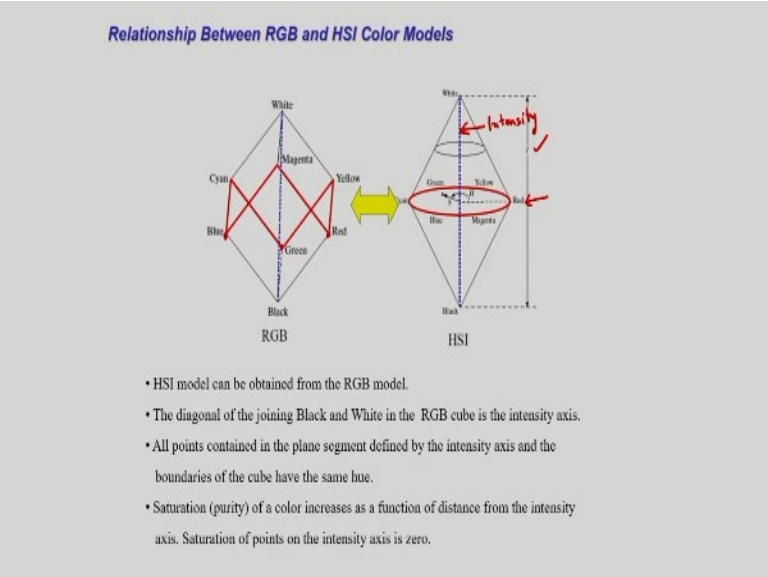
(Refer Slide Time: 6:03)



Now, based on this I have one color model that is the HSI color system, that is the hue, saturation and the intensity. So, in this diagram you can see I am representing hue, saturation and intensity. So, you can see the intensity axis, that is the brightness. And hue is nothing but, this axis the red axis, this reference axis is the red axis, and this angle if you see, this angle, this angle corresponds to hue.

And if you see this length of this vector from this to this, this corresponds to saturation. So, saturation is 1 for a pure color and less than 1 for an impure color, and here I have shown three components, one is the intensity component that is nothing but the brightness, hue means the color, and saturation means the purity of color.

(Refer Slide Time: 6:56)

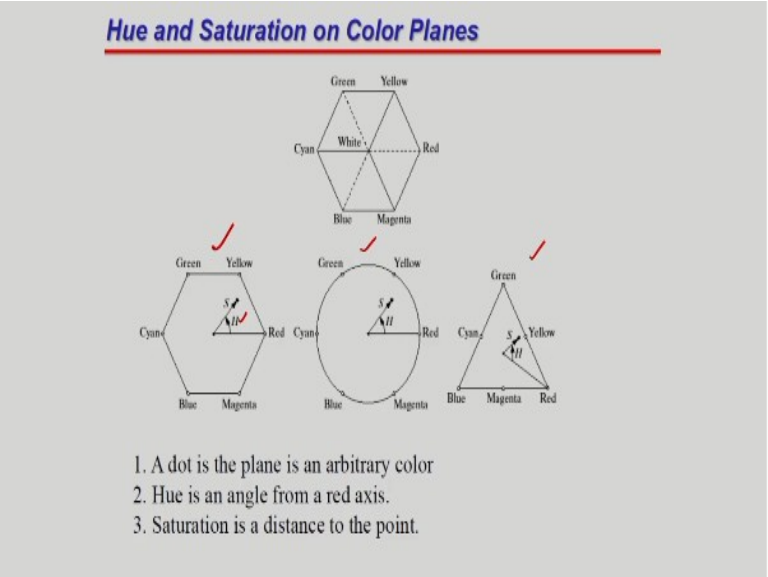


And in this figure what I have shown, the relationship between RGB and the HSI color model. So, first one is the RGB color cube, the second one is the HSI model I am showing like this. So, what you can see, in the RGB model you can see all colors, the primary colors, the red color and the green color and the blue color, these are the primary colors, also you have seen the secondary colors cyan, magenta and the green. And corresponding to this RGB cube, you can see the intensity axis that is connecting from the black point to the white point, that is the intensity axis.

And in the second figure you can see that is the HSI model. In this case also I am showing the intensity axis connecting the black point and the white point. So, if you see here, this is the intensity axis. And after this I am considering the hue and saturation with respect to the reference axis, the reference axis is the red axis, this is the reference axis, red.

So, hue means the angle with respect to the red axis saturation means the length of the vector. So, if you see here, the saturation of a color increases as a function of distance from the intensity axis and saturation of points on the intensity axis is 0. So, if I consider saturation of the points in the intensity axis, so this axis is, this is the axis, that is the intensity axis. So the saturation of the points on the intensity axis is 0. So, this is about the relationship between RGB and HSI color model.

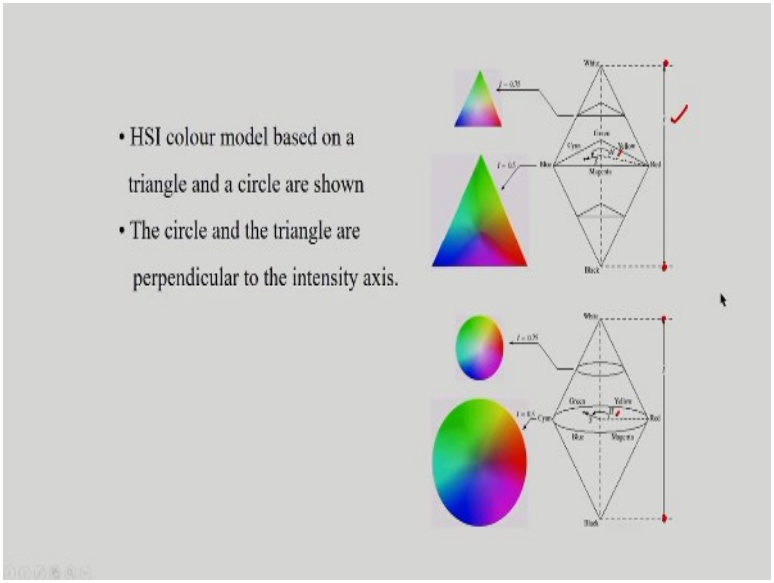
(Refer Slide Time: 8:45)



And in this case I have shown some models to represent hue and saturation on color planes. The first one is the hexagon color model, the second one is the circular model I am considering and the third one is the triangle model. So, the concept is same, and axis, the reference axis is red and I am considering the angle with respect to the red axis, that is the hue and saturation is nothing but the length of the vector.

So, if you see this diagram, the saturation is a distance to the point, that is the length of the vector. Hue is an angle from that red axis, so I am considering this. So I can consider any one of these model, the hexagon I can consider, maybe circle I can consider or the triangle I can consider.

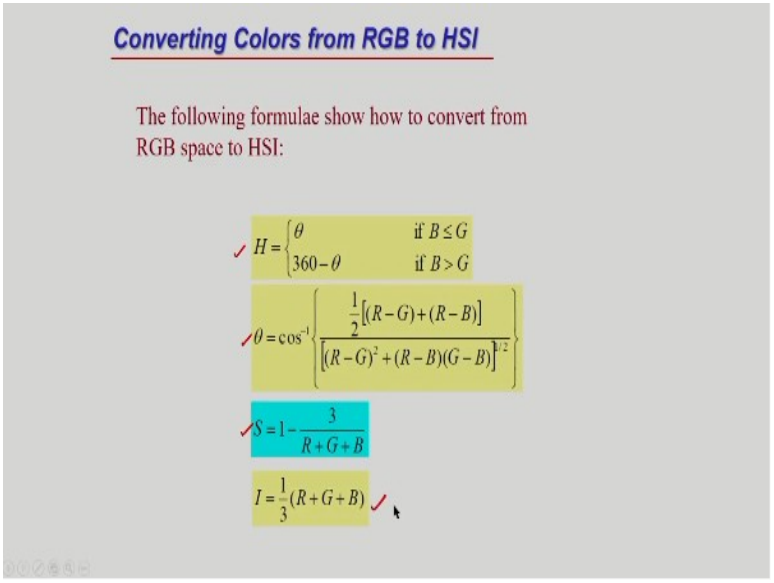
(Refer Slide Time: 9:40)



And in this case, I have shown here, again, the same thing the triangle model I am considering. And in this case if you see here, I have shown the intensity axis, that is connecting. This is intensity axis, connecting the black point and the white point. In the second case also, in the circle model I have shown the intensity axis connecting the black point and the white point, that is the intensity axis.

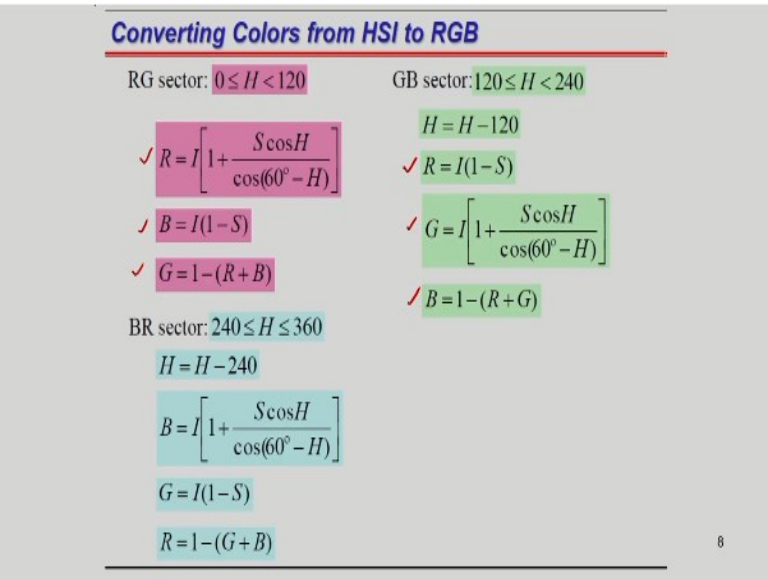
And corresponding to this, if you see the hue, what is hue? Hue is the angle with respect to the red axis. So, in both cases you can see the hue, this is the hue. And saturation is nothing but the length of the vector, if it is 1 then saturation will be 1. That means, I am getting the pure color. Corresponding to the points in the intensity axis the saturation will be 0, that means it is not the perfect color. So, this is about a representation of the HSI color model.

(Refer Slide Time: 10:41)



And some conversion formulas. Here you can see the HSI can be converted into RGB and also RGB can be converted into HSI. So, in this case I am considering that converting colors from RGB to HSI. So, I have the RGB values, from the RGB values you can determine the H, H is the hue and that means, in theta actually I can represent, the hue is theta and saturation you can see saturation is represented like this, and the intensity is nothing but 1 by 3 R plus G plus B. I think you did not remember all these formulas, only this the last formula is important that intensity is nothing but 1 by 3 plus R plus G plus B. So, this is the conversion from formula from RGB to HSI.

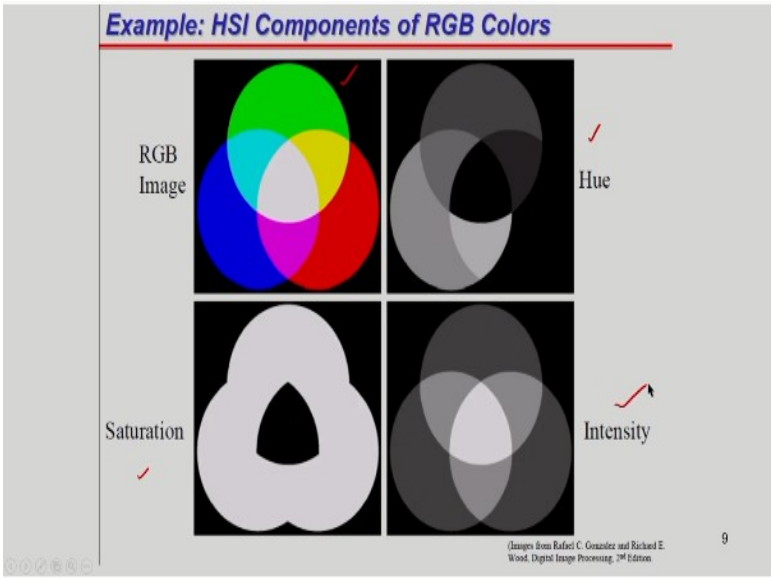
(Refer Slide Time: 11:28)



Similarly, I can convert the HSI to RGB. So, you can see all these formulas. And corresponding to RG sector, that is H lies between 0 to 120 degree, corresponding to this you can see the R value, G value and the B value. Similarly, corresponding to the GB sector, that is H lies between 120 degree to 240 degree, and corresponding to this you can see the values, the R value, G value and the B value. And similarly, for the BR sector you can get the values, R value, G value and B value.

Again, I think you need not remember all these equations. For exam what you can do, for exam these equations will be provided, only you have to find the values, you can convert the HSI into RGB or RGB into HSI.

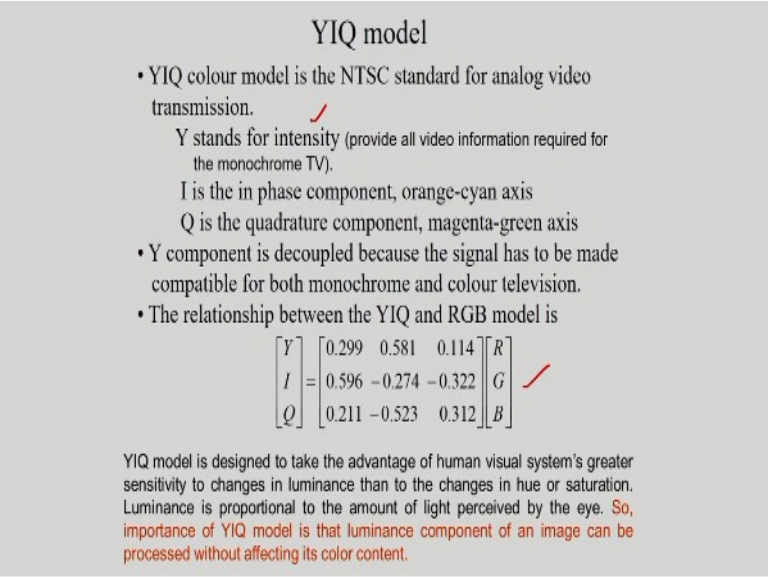
(Refer Slide Time: 12:25)



In this example, I have shown the HSI components of RGB color. The first one you can see, the RGB image I have shown here, the first figure is the RGB image I am showing here. The second I am showing the hue component of the image, the second is the hue component. The third one is the saturation component.

So, you can see, this portion is the black, black means because the maximum white light, that means the saturation will be 0, saturation means the purity of the color, that means the amount of white light added with a particular color. So, that is why the situation is 0, corresponding to the white portion and you can see the intensity values here. So, I am showing the HSI components of the RGB colors.

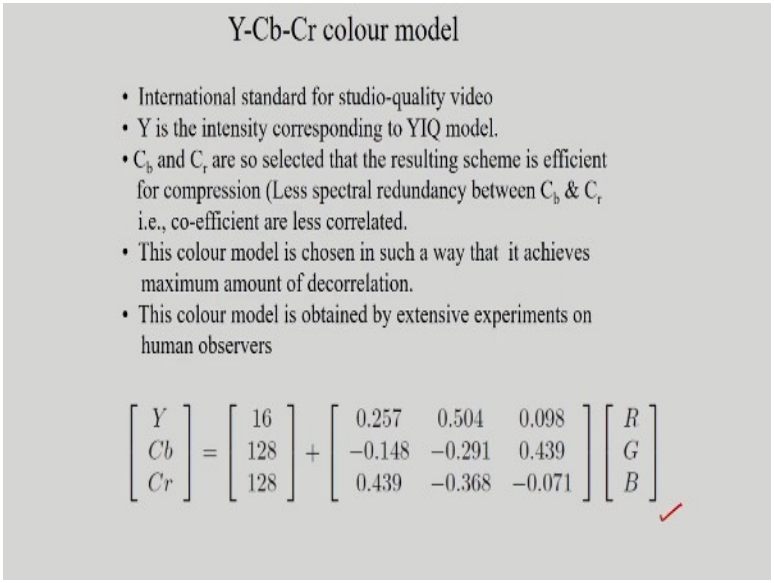
(Refer Slide Time 13:14)



The next model is YIQ model, that is mainly used in NTSC standard analog video transmission. In this case, Y stands for intensity component and I is the in phase component and Q is the quadrature component. And I have shown that conversion formula from RGB to YIQ. So, by using this equation you can convert the RGB value into YIQ value. And in this case, the same concept, because Y stands for intensity.

So, that means the importance of YIQ model is there, luminance component of an image can be processed without affecting the color components. So, mainly I can separate the brightness or the intensity from the color information. The color information mean, in this case, I and Q, the in phase component and the quadrature component. That is I am separating the luminance from chrominance.

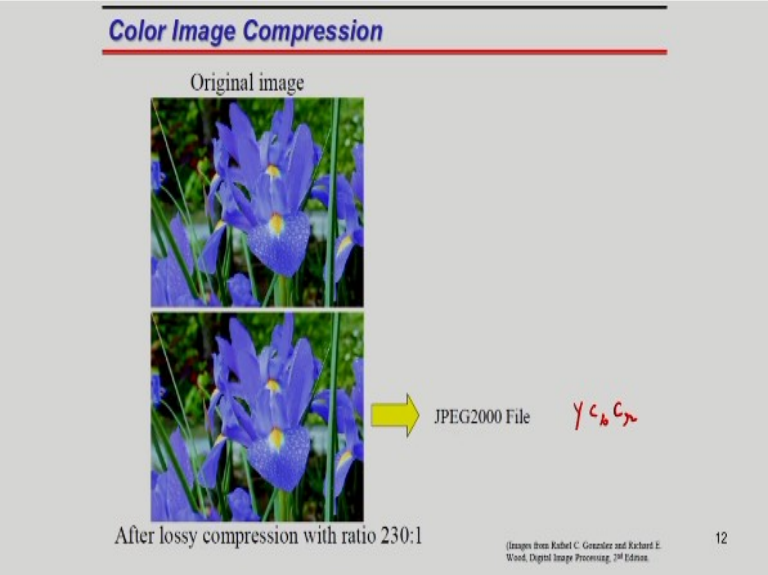
(Refer Slide Time: 14:18)



The another important model is Y Cb CR color model and this is mainly used in color image compression. So, Y is the intensity component corresponding to YIQ model and Cb and Cr are the color components, and it is mainly used for the color image compression.

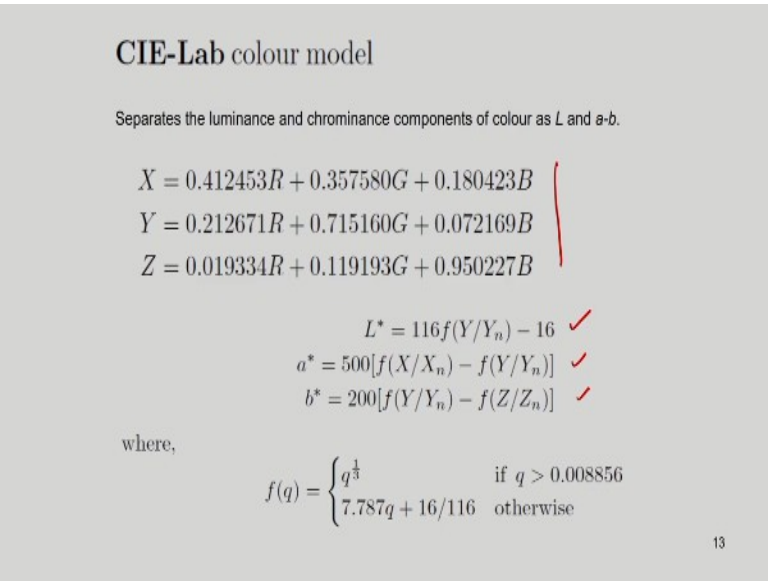
Why it is used? Because the spectral redundancy between Cb and Cr, that means, coefficients are less correlated, the Cb and Cr are less correlated. Because of this, this model, the Y-Cb-Cr model is used for color image compression. And here I have shown the conversion formula. So, you can convert the RGB values into Y Cb Cr, and by using this conversion formula you can do this.

(Refer Slide Time: 15:12)



So, here I have shown one example of color image compression. So, I have the original image and I am applying to JPEG2000. So, in this case, the color model is used Y Cb Cr. So, this color model is used for this compression and you can see the image after the compression.

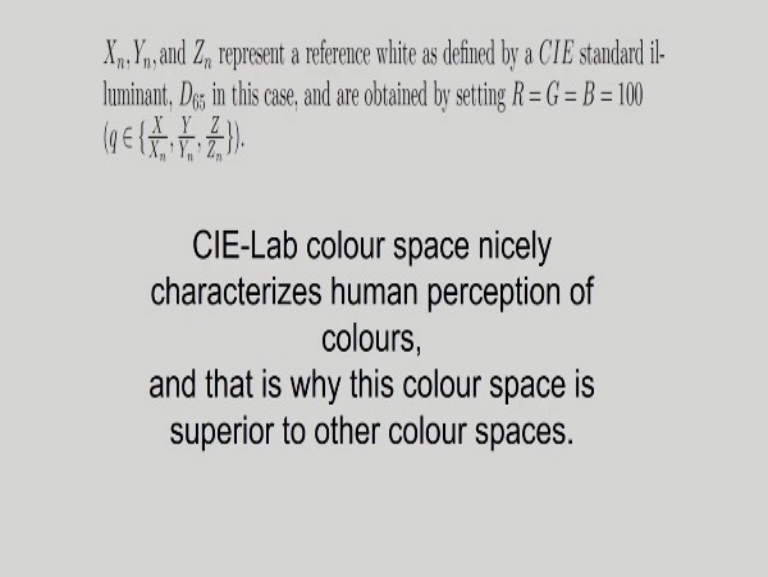
(Refer Slide Time: 15:38)



And finally, I want to show another important color model that is the CIE Lab color model. In many computer vision applications this model is used. So, already you know this conversion

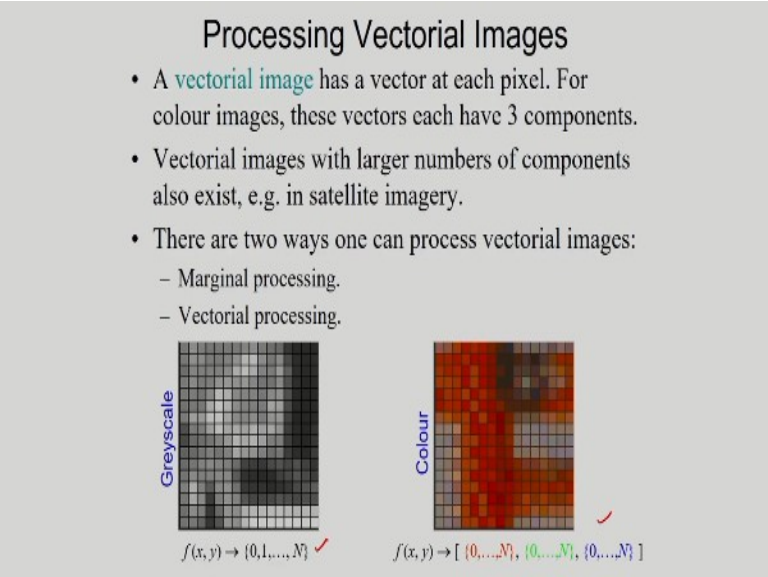
formula that is how to convert the RGB into X, Y, Z system. And based on this you can calculate the L value, a value and the b value by using this conversion formula. You can see all these conversion formula and you can implement this one so the RGB can be converted into Lab color model.

(Refer Slide Time: 16:14)



And in this one important property is the Lab color space nicely characterizes human perception of colors, and that is why this color space is superior to other color spaces. So, in many computer vision applications like skin color segmentation, background modeling, in video surveillance, so this color model, that is the Lab color model is applied.

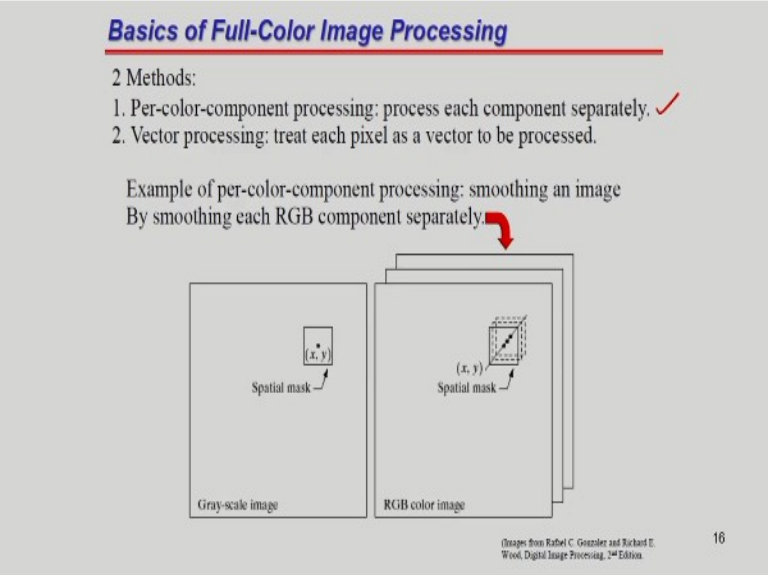
(Refer Slide Time: 16:40)



And already I have explained the concept of the color image processing. One is the marginal processing, another one is the vector processing. So, first figure if you see that is I am considering the grayscale image. The second one is I am considering the color image. So, corresponding to the color image I have to consider the vector pixel, corresponding to a particular pixel I have three components, one is the R component, G component and the blue component.

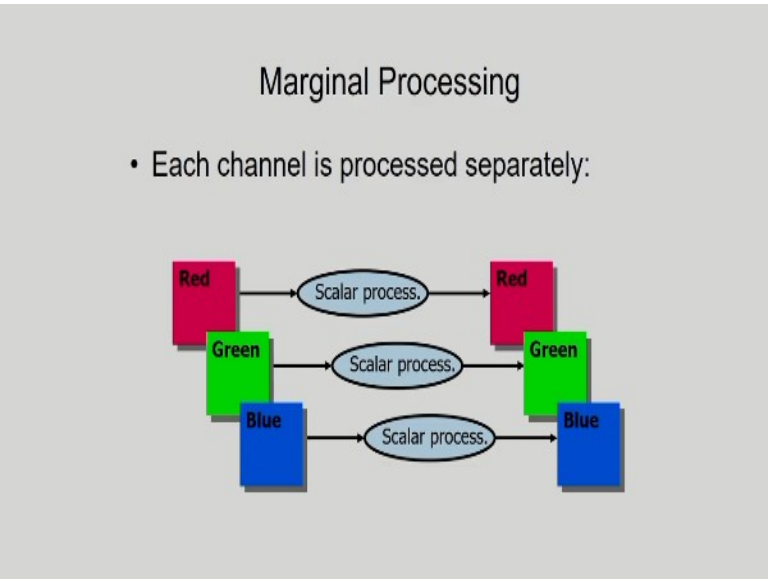
So, in the marginal processing what I can do, I can process the RGB components separately. But in case of the vector processing, I can consider the vector pixel as a whole, that means, I can consider RGB as a whole, as a vector pixel, and I can process all these components together, that is the vector processing. So, in the next slide, I can show the difference between the marginal processing and the vector processing.

(Refer Slide Time: 17:41)



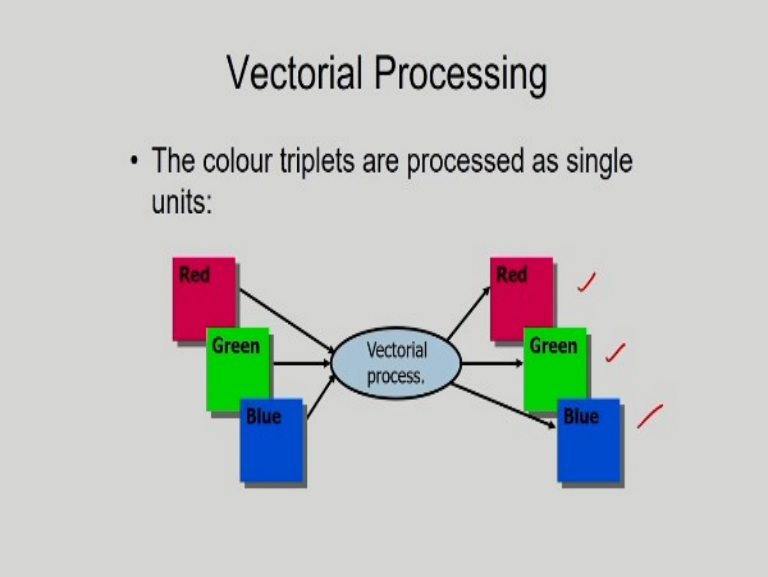
Here you can see I have shown that grayscale image and here I have applied the spatial masking operation. In the second case, I am showing the RGB color model and again in this case I am applying the spatial mask. So, in case of the marginal processing, what I have to do, that is process the RGB components separately. So that is the marginal processing.

(Refer Slide Time: 18:17)



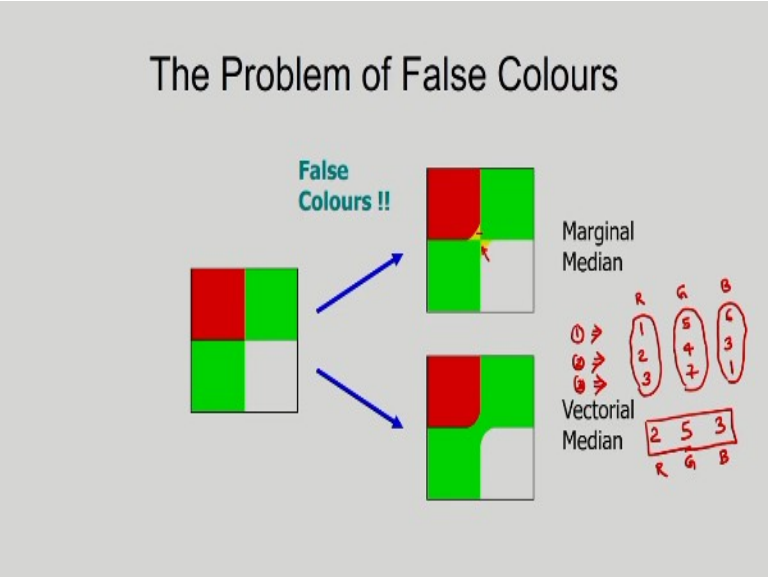
In case of the vector processing, treat each pixel as a vector to be process. In this figure, I have shown the concept of the marginal processing. So, my channel is red channel, green channel and the blue channel. So, red is process separately, green is processed separately and the blue is processed separately. So, that is called a marginal processing.

(Refer Slide Time: 18:37)



In the second example, this example I have shown the vector processing. So, I can consider RGB as a single unit, that is the vector pixel. And I can do the processing together, that is the processing for the R component, processing for the green component and processing for the blue component together and after the processing I am getting the output, output is the R, G and B. This is called vector processing.

(Refer Slide Time: 19:06)



Now, in this example, I have shown the difference between the marginal processing and the vector processing. So, one example I am considering that is the marginal median, the median value I am considering and second one I am considering the vector median.

So, in this case you can see in case of the marginal median, if I want to determine the median of the pixels. So what will happen? You can see the color, the yellow color that color is appearing here, that is not originally present in the image. In the original image this color was not present. That means, the color distortions take place. But in the vector median I am getting the original colors.

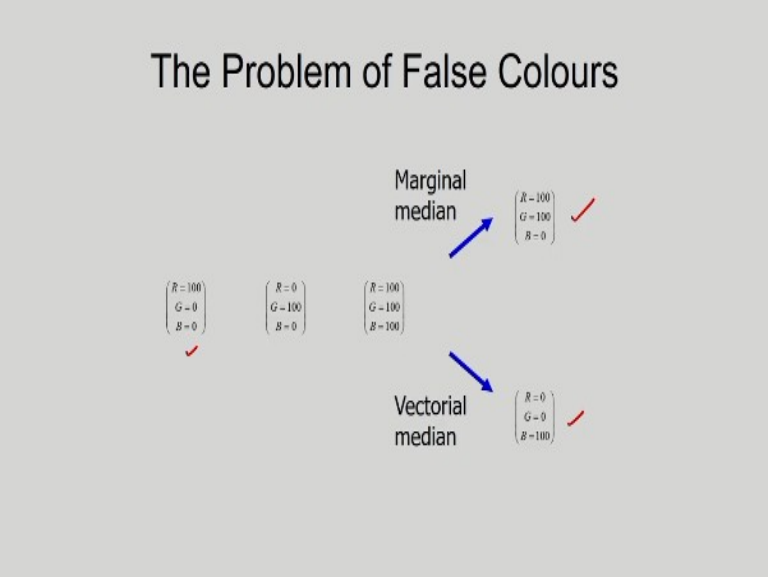
Let me give one example. Suppose, I have the R value, G value and B value, and I am considering suppose three pixel, the first pixel is number 1 pixel, the values are suppose 1, 5, 6. Second pixel I am considering, the second pixel is suppose 2, 4, 3. Third pixel I am considering suppose 3, 7, 1. So, I have three pixels, for the first pixel RGB values are 1, 5, 6, for the second the pixel the RGB values are to 2, 4, 3, and for the third pixel the RGB values are 3, 7, 1.

So, in this case, if I want to determine the median for this, the marginal median, that means for this channel I have to determine the median. So, what will be the median? The median will be 2. Corresponding to the green channel I am determining the median value, the median value will be

5, and corresponding to the third channel I am determining the median value, the median value will be 3, that is the marginal medium I am doing.

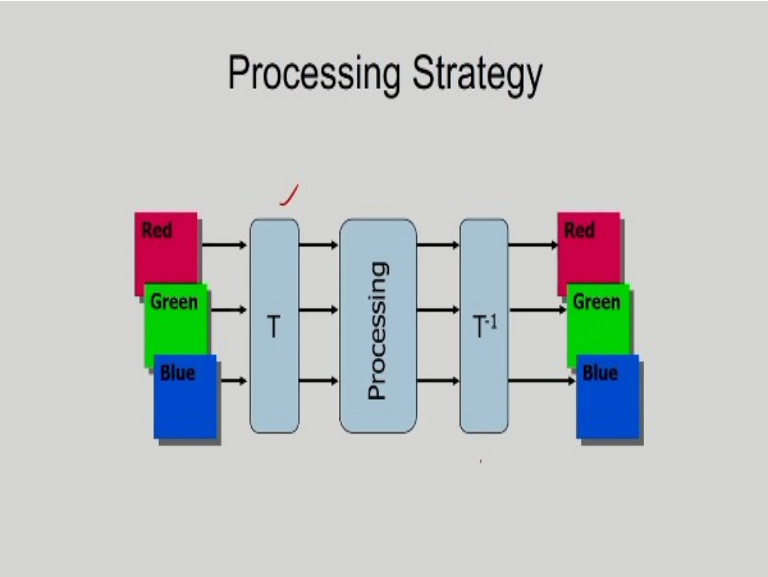
And corresponding to this what you are getting, you are getting the value 2, 5, 3. These 2, 5, 3 pixel value, that is the R is equal to 2, G is equal to 5 and B is equal to 3, this 2, 5, 3 is not available in the original image, that is why I am getting some other color in the image, the output image you can see that color, yellow color I am getting because of the marginal medium. This pixel 2, 5, 3 is not present in the original image. You can understand the concept of the marginal median, so what is the problem with the marginal median. That is the false color, that is the color distortions take place.

(Refer Slide Time: 22:03)



I am showing the same example again. So, I am considering three pixels, the first pixel is this, the value is R 100, 0, 0; the second is 0, 100, 0; the third one is 100, 100, 100. So, I can determine the marginal median, so marginal median will be 100, 100, 0, but vector median I can apply some algorithm. So, vector median may be something like this. So I will explain how to determine the vector median. So, in this case you can see this 100, 100, 0, that is not available in the original image that pixel values, that is nothing but the false color, the color distortions.

(Refer Slide Time: 22:45)

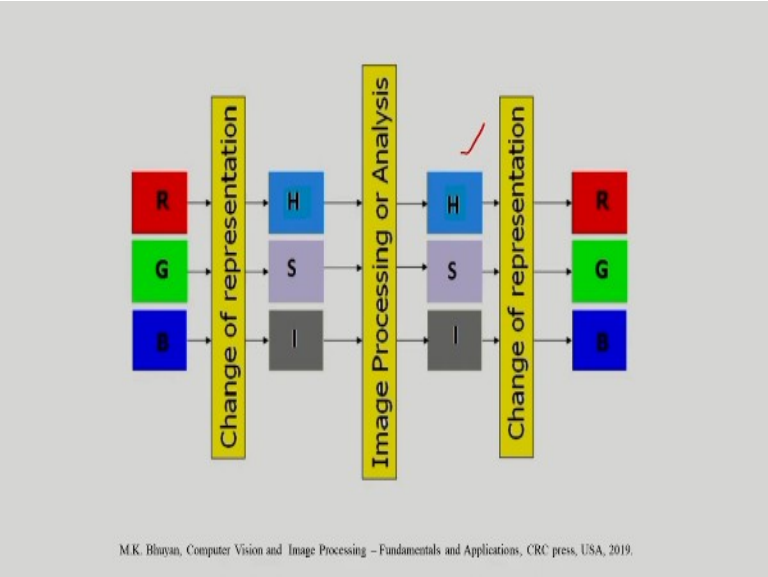


So, for color image processing what I can do, you can see here. I have the image, so I have the vector pixel RGB. So, I can apply transformation, so T is the transformation. And after doing this transformation I can do the processing, and after this I can do the inverse transformation to get the RGB value.

So, what is this transformation? Suppose, the transformation may be the RGB values I can convert into HIS, the hue, saturation and the intensity. And after this I can process only the intensity component or maybe I can process intensity component and the color components separately, that is the processing.

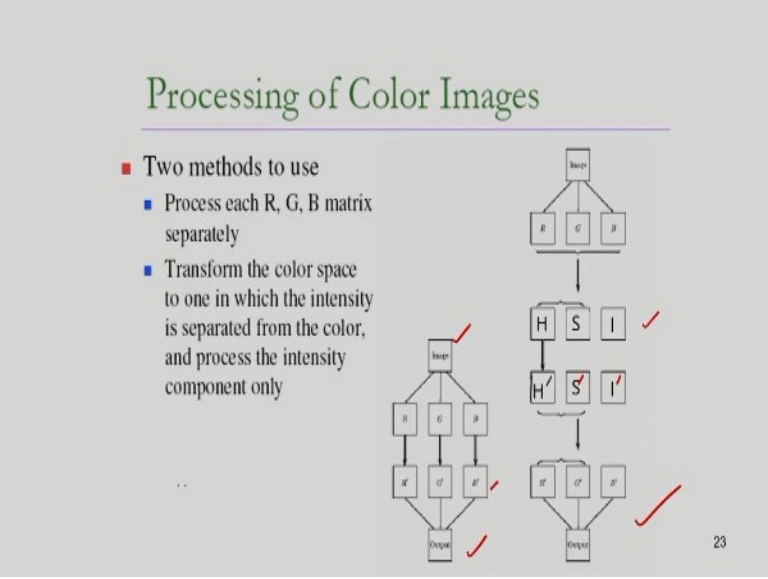
After this I have to do inverse transformation, that means the process HSI value is converted into RGB, that is the processing technique.

(Refer Slide Time: 23:41)



Here the same thing I am showing here, the RGB value what I am converting, I am converting into HIS. After this I am doing the processing, maybe I can only select the intensity component or maybe I can do the processing separately, I component and saturation component and the hue component. After processing I am getting HSI and after this we have to do inverse transformation, that is HSI is converted back to RGB. That is the processing technique.

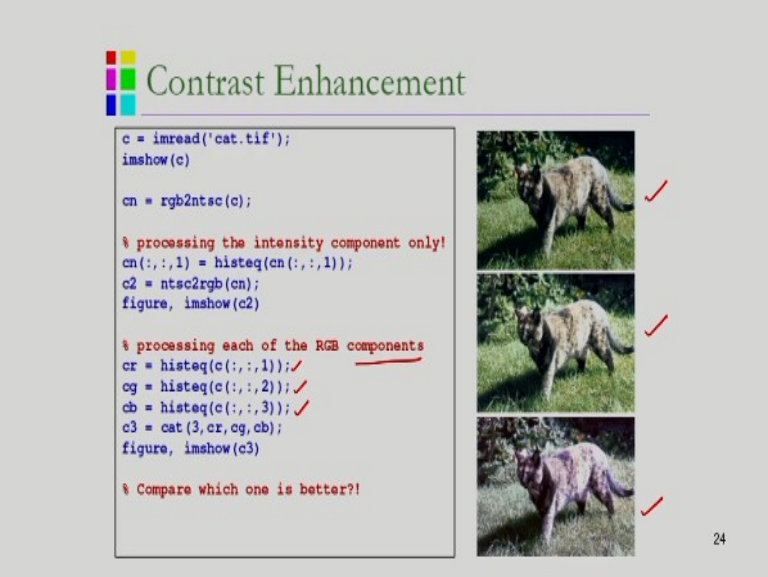
(Refer Slide Time: 24:17)



In this case, again, I am showing the same example. In the first figure if you see, in the first figure what I am doing, corresponding to this input image I have the RGB components. And in this case, without converting RGB into HSI, I can do the processing for R component, G component and the blue component. So, after processing I am getting R dash, G dash and B dash, so I am getting the output image like this.

But in the second case, if you see the second case what I am doing, from the input image I have the RGB components. So RGB components and that I am converting into HSI components, and after this I can do the processing, so I am getting H dash, maybe the S dash or the I dash I will be getting. And after this, I can convert the HSI into R component, G and B corresponding to the output image.

(Refer Slide Time: 25:18)

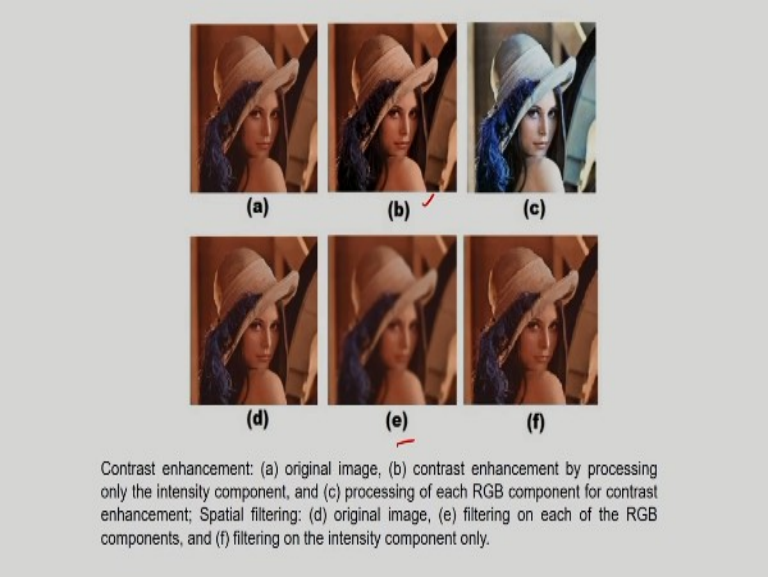


Here I have shown some examples, the contrast enhancement. So first, you can see the input image, input image is this. And what I am doing, first I am considering the contrast enhancement and for this I am only considering the intensity component I am considering, and I am applying the histogram equalization technique. So, corresponding to this I have a second image. Now, this is the second image I having.

In the second case, you can see what I am considering, processing each of the RGB components separately, that is the second case. So I am doing the histogram equalization for R component, G

component and the blue component, and corresponding to this I have this image. You can see the difference between these two images, the image number two and the image number three. In the image number two, what I am considering, only I am considering the intensity component of the image, that is the color value. And in the second case what I am considering, I am considering R value, G value and the blue value and processing separately.

(Refer Slide Time: 26:26)

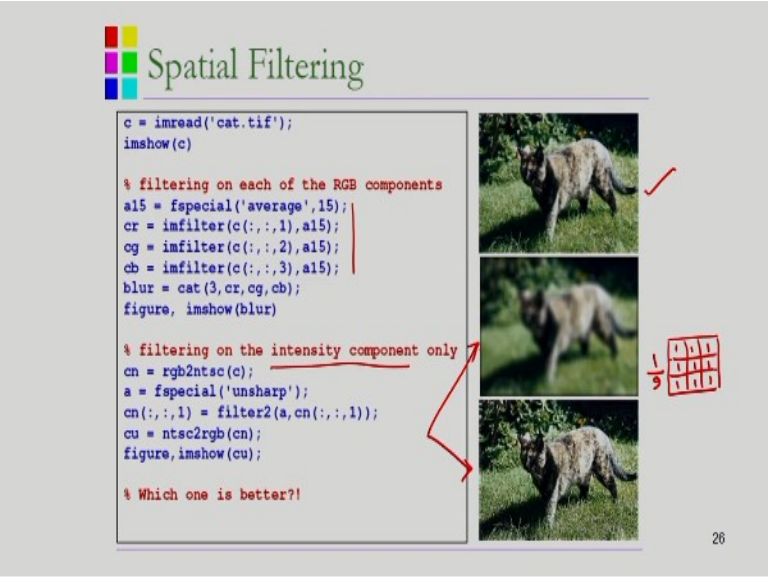


Contrast enhancement: (a) original image, (b) contrast enhancement by processing only the intensity component, and (c) processing of each RGB component for contrast enhancement; Spatial filtering: (d) original image, (e) filtering on each of the RGB components, and (f) filtering on the intensity component only.

In this case, I am showing the contrast enhancement. The first case if you see, a is the original image, b is the contrast enhancement by processing only the intensity component, so in the this case I am considering only the intensity component. The third case is the processing of each RGB component for contrast enhancement.

The second example if you see, I am doing the spatial filtering. So d is the original image, e is the filtering on each of the RGB components, that means e is the filtering on each of the RGB components, that is separately I am doing. And last one is filtering on the intensity component only, that I am considering. You can see the difference between these two cases.

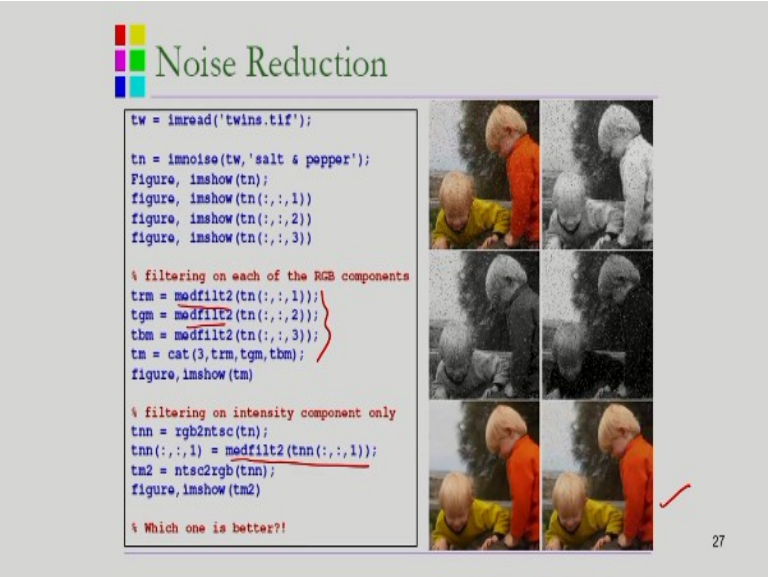
(Refer Slide Time: 27:18)



Again, I am considering another example, the spatial filtering. So, in this case what I am considering? I am considering the input image, the first one is the input image. And if you see the second image, what is the second image? I am doing the filtering. And in this case what I am considering, filtering on each of the RGB components separately, that means I am considering the R component, G component and the blue component separately.

In the second case, where I m considering I am only considering the intensity component and after this I am applying the spatial filtering. The spatial filtering already I have explained, the spatial filtering suppose if I apply the averaging filter, so what will be the mask? The mask is nothing but 1, 1, 1, 1, 1, 1, 1, 1, 1. So, this is the mask for the average filter. So, this mask I can apply only for the intensity component and you can see the difference between this image and this image, the second image and the third image.

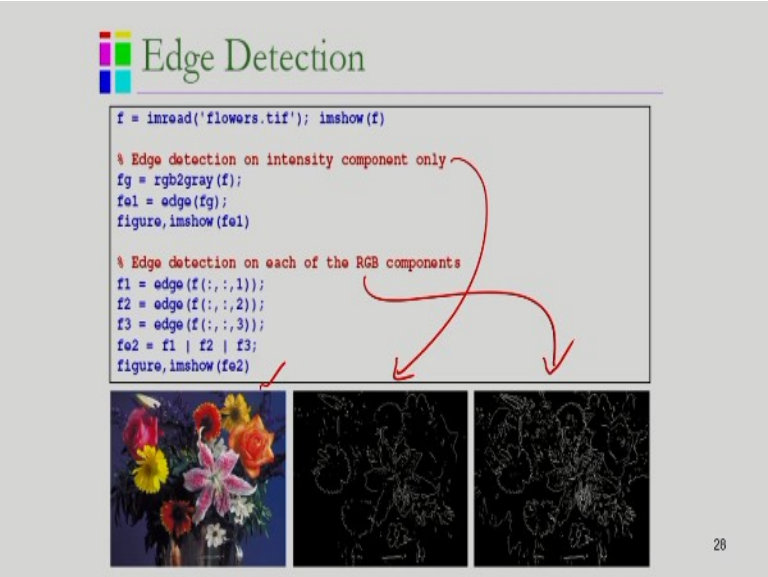
(Refer Slide Time: 28:25)



The next example I am showing that is I am considering the salt and pepper noise. And after this what I am considering, the filtering on each of the RGB components I am doing so, I am applying the median filters, you can see the median filters here. So, I am applying the median filters to remove the salt and pepper noise.

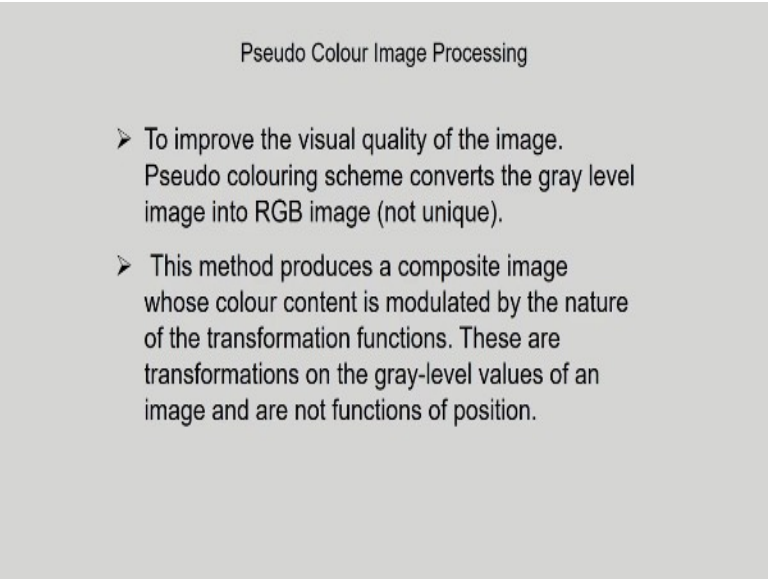
And in the second case what I am considering, I am considering the filtering only the intensity component, that means, I am applying the median filter only for the intensity component and you can see the output, the output is this. In the first case what I am considering, the filtering on each of the RGB components and I am applying the median filter to remove salt and pepper noise.

(Refer Slide Time: 29:10)



And this edge detection, I am going to discuss in my next classes the edge detection, but here I have shown one example. So, my input image is this and what I am considering, in the first case I am only considering the intensity component for edge detection and corresponding to this my output is this. In the second case what I am considering, edge detection on each of the RGB components and corresponding to this my output is this. So, I have shown these examples to explain the concept of color image processing.

(Refer Slide Time: 29:49)

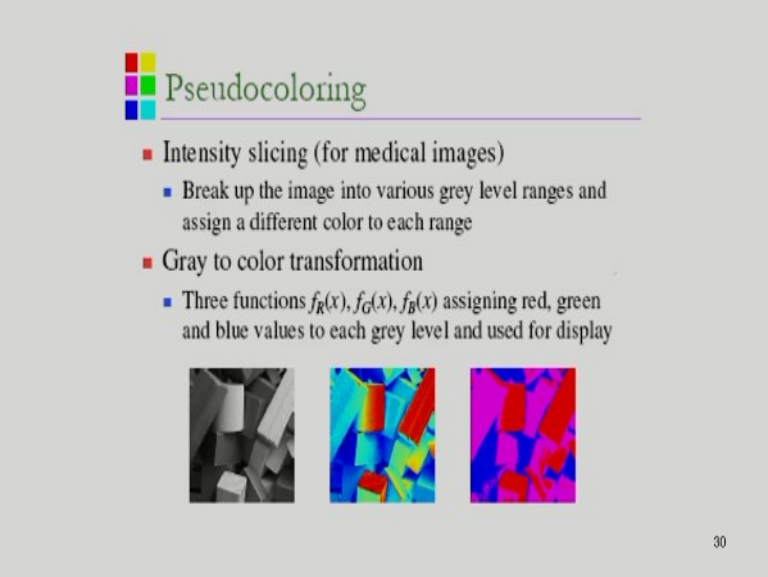


The next point is the pseudo color image processing, that is the false color processing. That is what is the false color? That means I can convert that grayscale value into the color value. So, for this I have to define some transformation and based on these transformation functions, I can convert the grayscale value into the color value.

Now, one point is important, that these transformation, these are the transformation on the gray level values of an image, but they are not functions of positions. And this method produces a composite image whose content is modulated by the nature of the transformation functions.

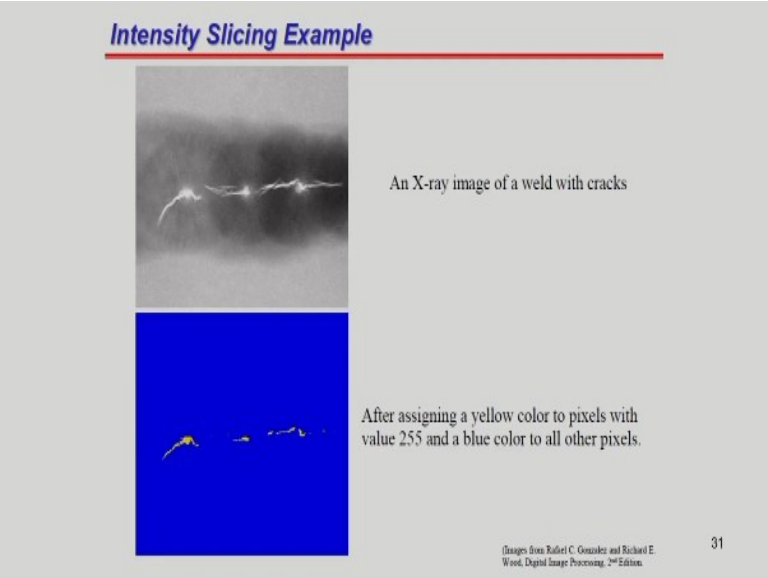
So, I have to consider some transformation functions and what is the method? So, this method produces a composite image whose content is modulated by the nature of the transformation functions and one point is important. So, these are the transformation on the gray level values of an image, but they are not a function of positions.

(Refer Slide Time: 30:57)



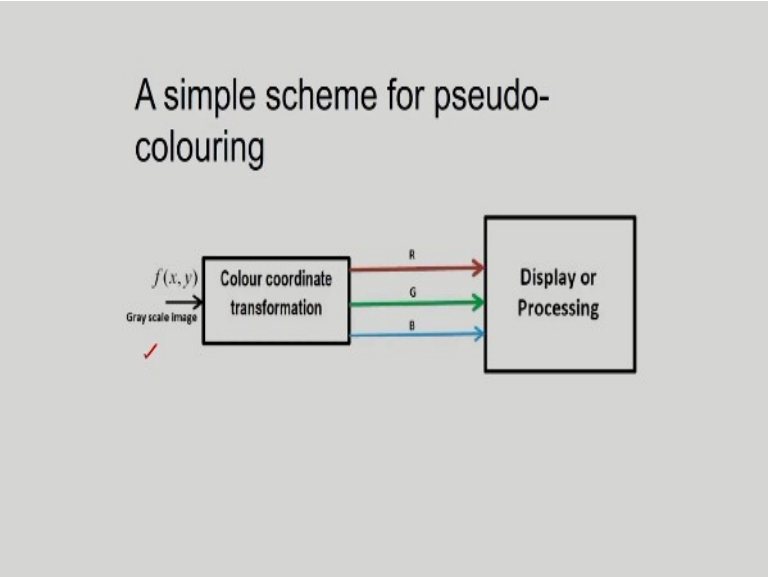
And here you can see, I have to define some transformation for the R component, G component and the blue component. And that means I have to assign the red color, green color, blue color corresponding to gray level pixel intensity values. So, this is nothing but the pseudo coloring, pseudo color means the false coloring.

(Refer Slide Time: 31:18)



So here you can see the example. So I have the grayscale image, and I am converting into the color image. So, here you can see the procedure is something like intensity slicing I am doing. So assigning a yellow color to pixels with value 255 and a blue color to all other pixels.

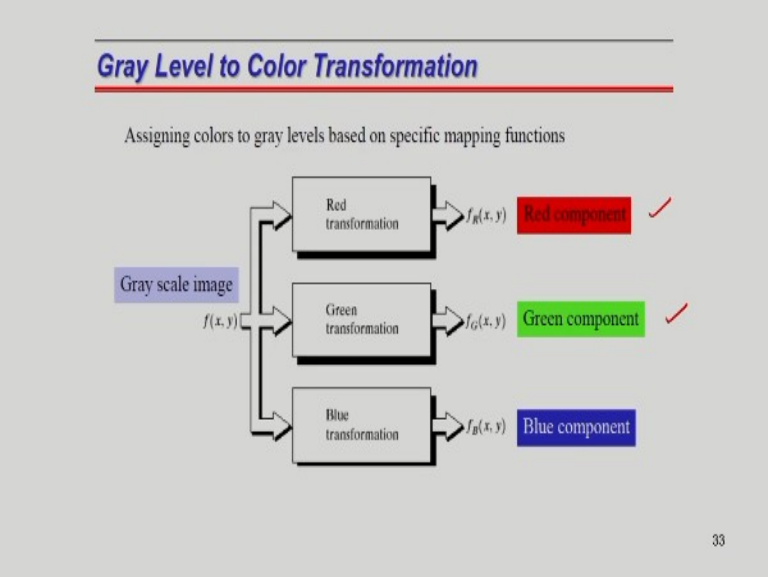
(Refer Slide Time: 31:40)



Now, in case of the pseudo coloring, so already I have explained about the transformation. So you can see the input is the grayscale image f x, y is the image, and I am considering some color

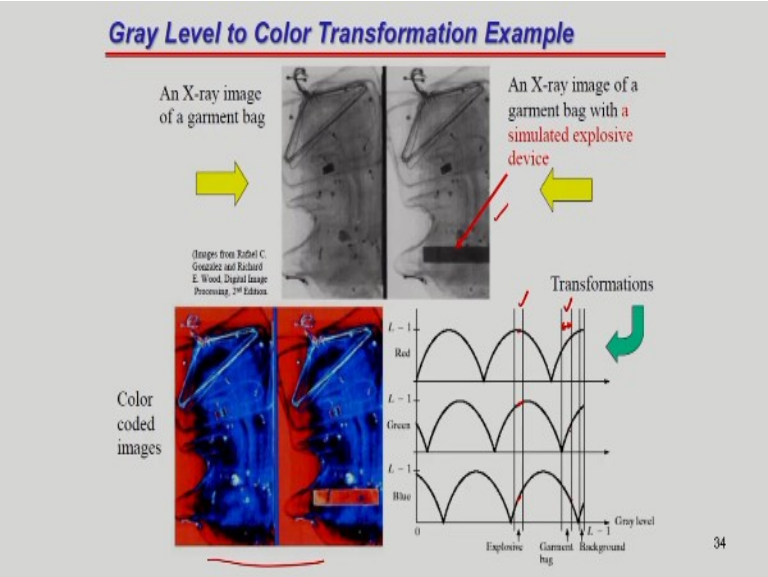
transformation. So color coordinate transformation I am considering. And after transformation, I am getting the R value, G value and the B value, and that I can display as a color image. So, I have to define the transformation, so how to define the transformation, in my next slide you can see.

(Refer Slide Time: 32:09)



So, I have the grayscale image value, grayscale image. And I am considering the transformation for the red component, transformation for the green component, transformation for the blue component and corresponding to this I have the red component, a green component and a blue component. So, I have these components red, green and blue components. So, let us see what type of transformation we have to use.

(Refer Slide Time: 32:37)



So, one example you can see, the X-ray image of a bag suppose, here I am considering, and we have to assign colors. In this case, how to do this? So, if you see, the first one is I am considering the grayscale image and I am assigning the colors, the color coded image I am getting.

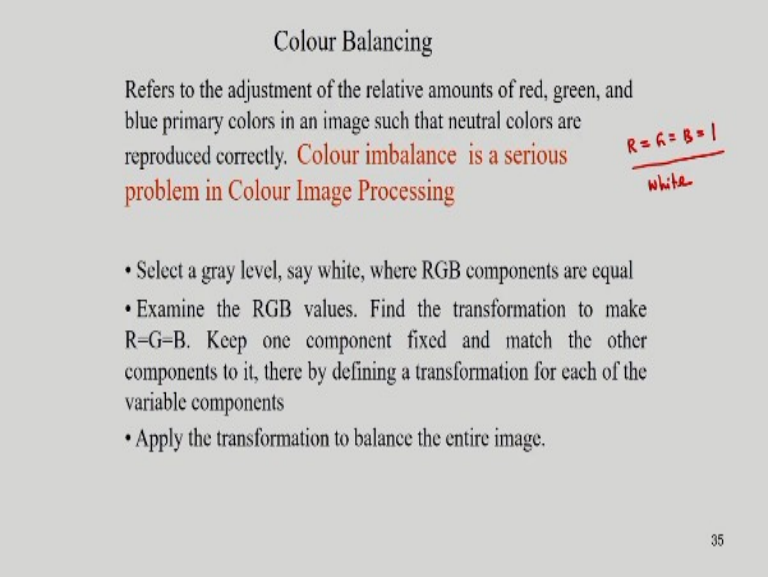
So for this I am considering some transformation, you can see the transformation, the transformation for the red component, green component and that blue component. And if you see these sinusoid, the frequency and the phase of the sinusoid are not same, then in this case corresponding to particular suppose this portion, I can assign a particular color based on the R value, based on the G value and based on the B value.

So, corresponding to this I can assign a particular color. Similarly, corresponding to this portion, suppose if I consider this portion, I have the R value, G value and the blue value, and corresponding to this I can define a particular color. So, you can see I am considering the transformation that means I am considering the sinusoid, but frequency and the phase it is different for the red component, green component and the blue component. So, that means changing the pace and the frequency of the sinusoid can emphasize in color ranges in the grayscale image.

Suppose, if all these transformation have the same phase, the phase is same suppose and the frequency is also same, then the output will be the monochromatic output. Because I will not be

getting the color image. That means, I am repeating this, suppose all these transformation will have the same phase and the frequency, then the output will be the monochromatic output. So, you can understand the concept of this transformation and by using this transformation, I can convert the grayscale image into color image.

(Refer Slide Time: 34:37)



Next important point is the color balancing. So, what is the color balancing? Generally if I take the image by a color camera, the digital camera, so the RGB output should be such that when mixed equally it should produce white color. That means, suppose if I consider R is equal to G is equal to B is equal to 1, and corresponding to this what I should get? I should get the white color. But actually, because of the sensor imperfections, I am not getting the actual color, that is called the color imperfection.

So, for this I have to do that color balancing. And any one of the component may become weak, but the components may be R component, G component and the blue component. So any one of the components may become weak and that is why the color mismatch will occur. So, that is the problem of color distortion, so for this I have to do color balancing.

So, one example is, in case of a digital camera generally the blue channel is noisy as compared to R and G component. So, in this case what is the procedure of the color balancing? Select the gray

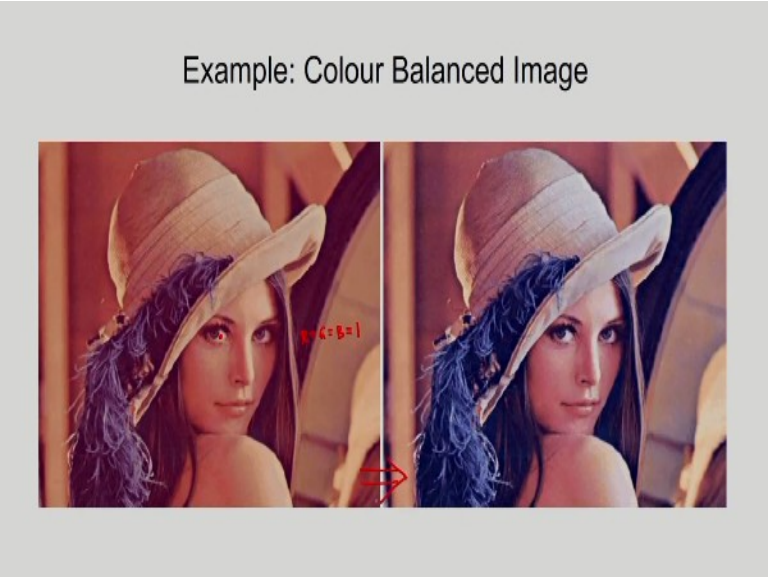
level, say suppose white, and in this case of the white point in an image R is equal to G is equal to B is equal to 1 that I know, corresponding to white.

And in this case, I have to find a transformation to make R is equal to G is equal to B because in the real case it is not equal but actually it should be equal for the white pixel, R should be equal to G, G should be equal to B for a perfect image. But in this case really it is not true. So, that is why I have to find a transformation to make R is equal to G is equal to B.

After this what I have to consider keep any one of these component fix and match the other components to it and by this we can define a transformation for each of the variable components. So, I can make R fix and I have to find the transformation for that green component, transformation for that blue component. So that R is equal to G is equal to B. And after this I have to apply this transformation to all the pixels of the image to balance the entire image. So, this is the procedure.

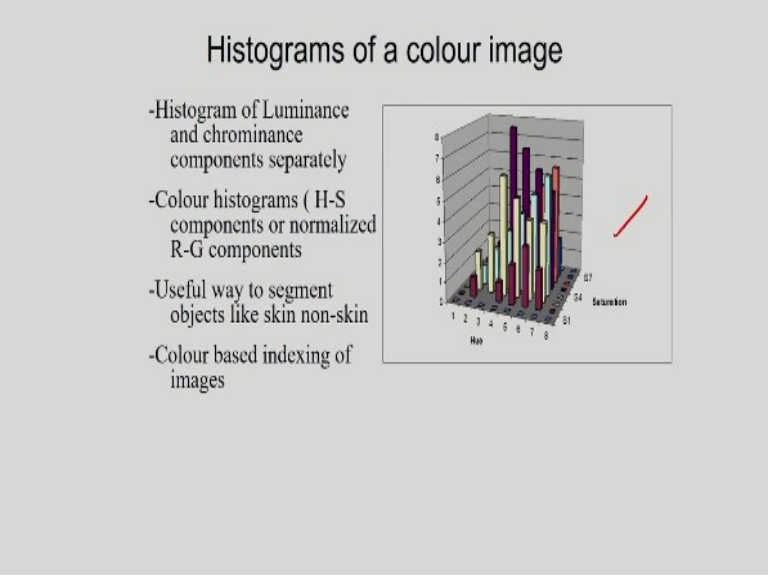
So, for a known pixel, that is suppose the known pixel is the white pixel, I know this condition that R is equal to G is equal to B is equal to 1, but actually because of the color imperfection, it is not true. So, what I have to do? I have to find a transformation to make R equal to G is equal to B. After this I can make one component fix, and what I have to do? I have to match the other components to it and for this I have to define a transformation. And I have to apply these transformation to all the pixels of the image to balance the entire image.

(Refer Slide Time: 37:49)



So, here I can show the example. So, corresponding to this, the known portion here, you can see the eyeball here. This portion is supposed white, corresponding to this pixel what I can do the R is equal to G is equal to B is equal to 1 that I know. And after this I am applying the color balancing technique and corresponding to this I am getting the output image. So, my output image is this, that is the color balance image.

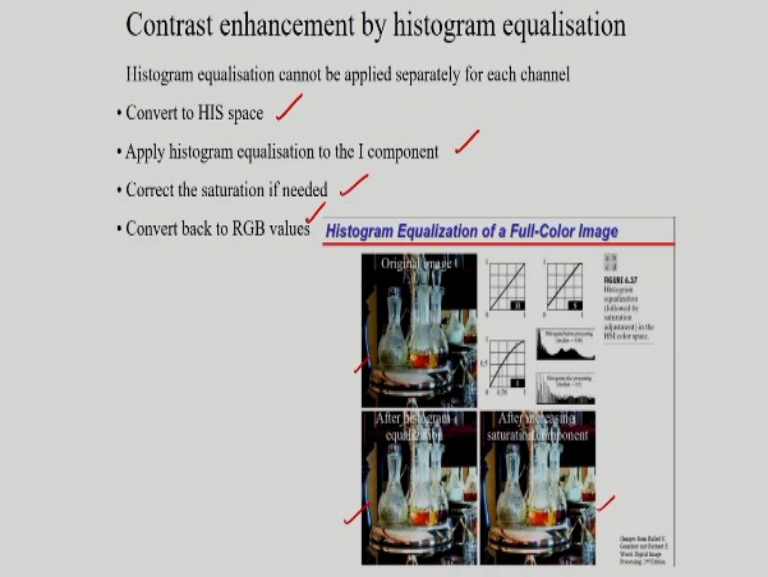
(Refer Slide Time: 38:14)



After this I can define the histogram of a color image. So, in this case what I can do, the histogram of luminance and chrominance components separately I can do, or otherwise that color histograms for the R component, G component and the B components also I can do.

And this color histogram is quite important, this color histogram I can consider as a feature, the image feature. So, for this what I can do, I can determine the histogram for the luminance component and the chrominance component. Chrominance components means the hue and saturation, like in this image.

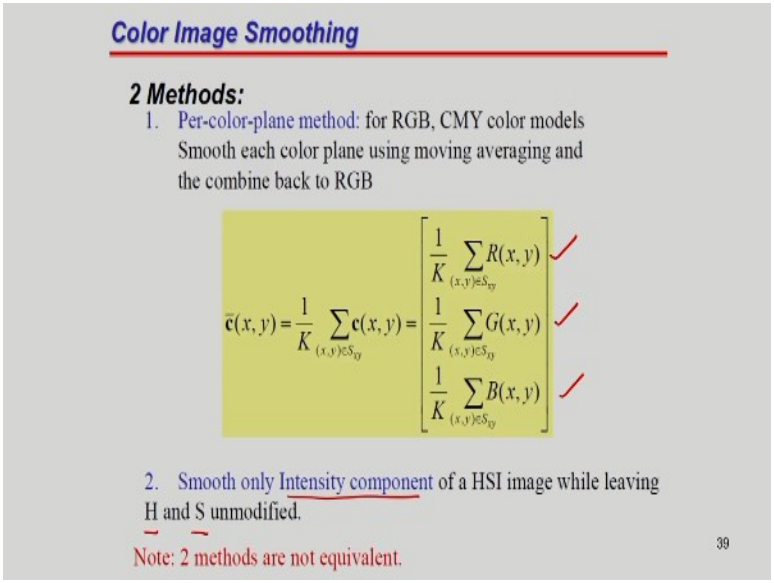
(Refer Slide Time: 38:48)



And in this case, I have given one example of the contrast enhancement by histogram equalization technique. So, here you can see I am applying the histogram equalization technique. So, first RGB is converted into HSI color space, this is the first step, after this I am applying the histogram equalization technique for the I component and maybe I have to do saturation correction, correct the saturation if needed and after this the HSI is converted back to RGB.

So, you can see I have the original image, after this I am applying the histogram equalization technique only for the I component, so that is my result. And after this I have to do saturation correction, so after saturation correction, this is my output image.

(Refer Slide Time: 39:36)



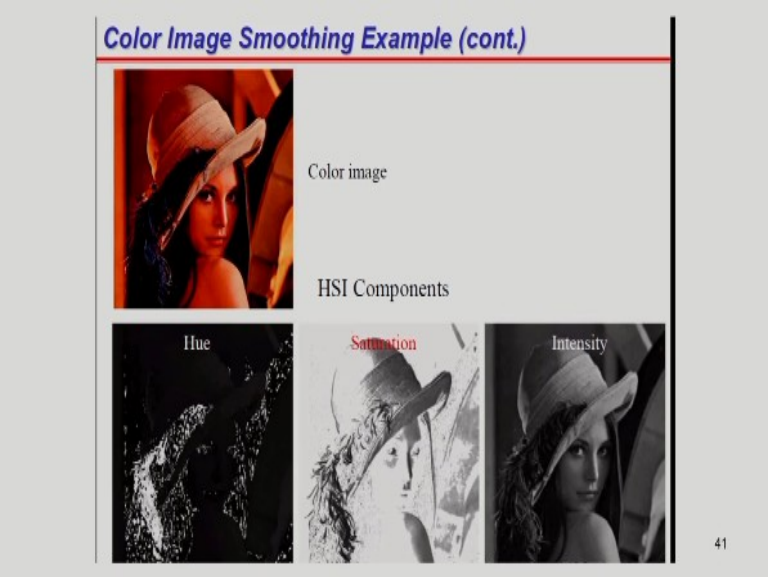
And the color image smoothing, so I can do the smoothing of the color image. So, in this case what I can do, so pre color plane method. So, for RGB, CMY color models, smooth easy color plane using moving embracing and combined back to RGB. So, you can see I am considering the moving average filter I am considering the R component, G component and the blue component for averaging.

In the second case, again, I am showing here, the second case what I am considering, I am only considering the intensity component for smoothing, that is the averaging but without considering the hue component and the saturation component. That means, I am only considering the intensity component, I am not considering the hue and the saturation component.

(Refer Slide Time: 40:27)

And in this case, you can see the example, the color image, the input is the color image. And I have shown that red image, the red color component, the green and the blue components of the image.

(Refer Slide Time: 40:41)



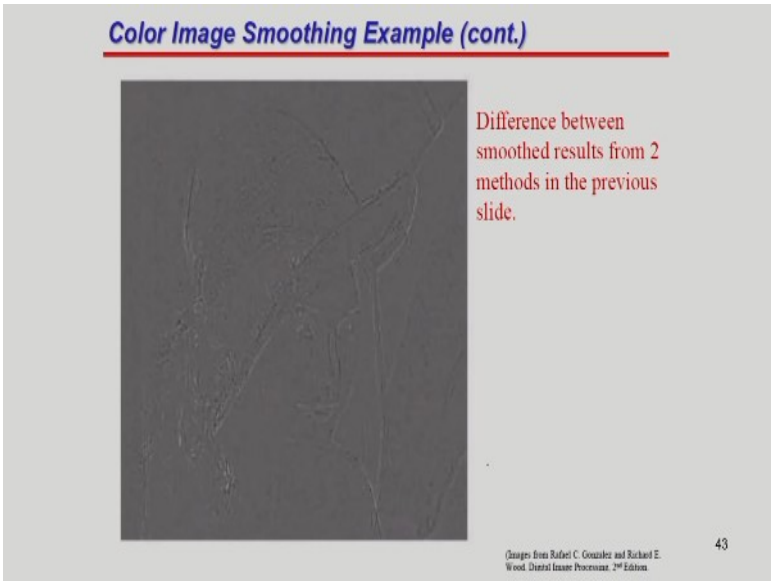
And in this case, you can see the color image and I have shown the hue component, saturation component and the intensity component that you can display, because the RGB can be converted

into HSI by using the transformation. So, I have given the equations. So, by using these equations, you can convert the color image into hue, saturation and intensity.

(Refer Slide Time: 41:04)

And in this case, I have shown the example of the color image smoothing, smooth all RGB components, the first case. The second case, only I am considering the intensity component and smooth only the I component of the HSI color model, you can see.

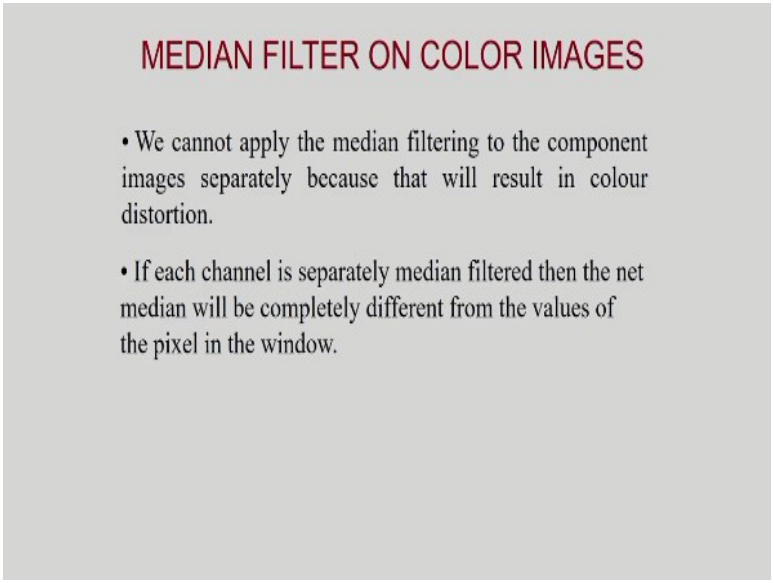
(Refer Slide Time: 41:22)



And this is the difference between these two images. One is the first image, the first image you can see, the first image is this that is I am considering the RGB components, the second image is only I am considering the intensity component. And the difference between these two I am getting this one, you can see the difference between these two.

And also, I can apply the high pass filter that is the color image sharpening. So, again I can do like this, so I can consider RGB components or I may consider only the intensity component for a HSI model. So, like already I have explained about this and this is the difference between the certain results from the two methods in the previous slide.

(Refer Slide Time: 42:07)

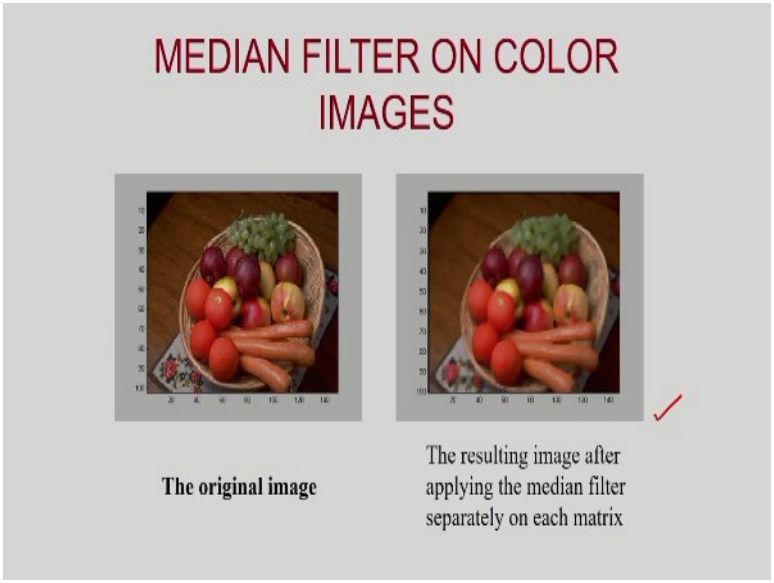


After this I will discuss the concept of the median filter, already I have explained the two techniques of color image processing, one is the marginal processing and another one is vector processing. And I have explained what is the problem with marginal median. So, if I determine the marginal median, then the color distortions take place.

So marginal median means, if I consider RGB separately and if I determine the median for the R component, G component and the blue components separately, then in this case the color distributions take place. So, that is why I cannot apply marginal median filter that is the scalar median filter.

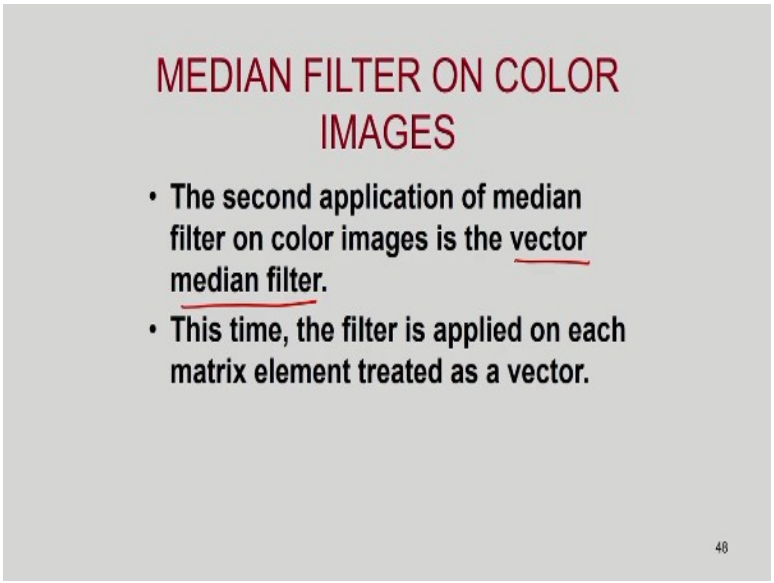
So, for this I have to consider the another technique, that is called the vector median filter. So for vector median filter we have to consider one window, and in this window I have to apply the vector median filter.

(Refer Slide Time: 43:12)



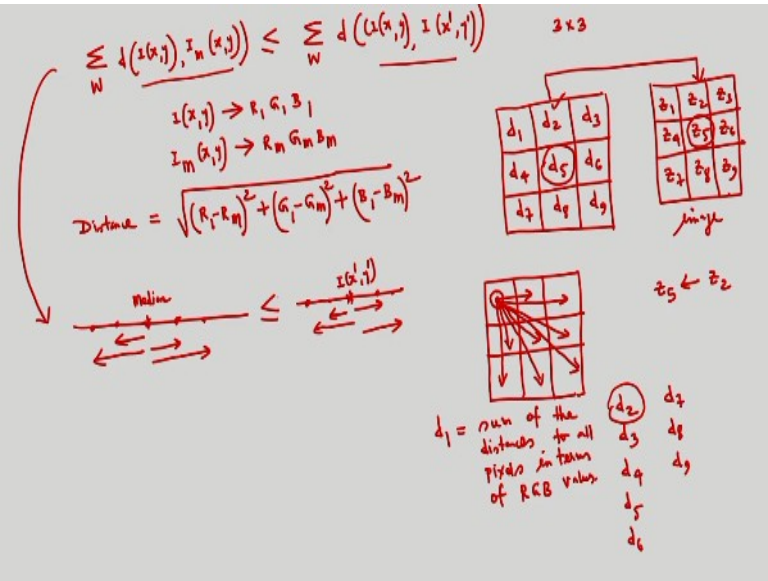
In this case you can see one example. I have shown the original image and after applying the scalar median filter, that is the marginal median filter, that means I am applying the median filter for the R component, G component and the B component separately. Then in this case the color distortions take place, then in this case you can see the output image corresponding to marginal median filter, the scalar median filter.

(Refer Slide Time: 43:43)



So, that is why we have to consider the concept of vector median filter. So, how to develop the vector median filter? So, what is the concept behind the vector median filter?

(Refer Slide Time: 43:56)



So, for vector median filter suppose, I want to find a distance between two pixels. So, what is this vector median, suppose the distance between two pixels in terms of RGB value, and I am considering one window and in this window I am finding the distances.

So, in this case I am considering the one pixel, the pixel is I x, y and corresponding to this pixel the RGB value is RGB value is R1, G1, B1 and I am considering, suppose the median filter, the vector median filter. In this case I am considering the median pixel, corresponding to a median pixel Im x, y, the R value, G value and the B value will be Rm, Gm and the Bm.

So, if you see this expression, what is the meaning of the median? So, if I find a distance between the pixel I x, y and the median pixel, that is the meaning of this, and summation over the window, so I have to consider all the distances and after this I have to take the summation corresponding to that particular window. And also I can determine the distance between two pixels, distance means, the distance in terms of RGB values and the same window I am considering.

Now, the distance between the pixel and the median, that will be less than or equal to the distance between any two pixels within this window. So, how to calculate the distance? The

distance in terms of the RGB value and this is the Euclidean distance I am considering, distance between any pixel I x y and the median pixel I am considering, and that distance will be less than equal to the distance between any two pixels. So, that is the motivation behind the median filter.

So that means, pictorially I can show like this. Suppose this is the median pixel, so I can find the distance between the median and other pixels, I can find and that will be less than or equal to. So if I consider another pixel and I am finding the distance between this and other pixels within this window, then in this case the distance between the pixel and the median pixel, any pixel and the median pixel will be less than or equal to distance between two pixels within that window. So, pictorially I have shown this one.

Now, so how to calculate vector median? So, for calculating the vector median, I can consider one window, maybe the symmetric window I am considering, the 3 by 3 window I can consider. Suppose I am considering a 3 by 3 window, so this 3 by 3 window is considered and I am calculating the distances d1, d2, d3, d4, d5, d6, d7, d8, d9.

Now, in this case, corresponding to this 3 by 3 window, my center pixel is this, this is my center pixel. So, if I consider this image suppose, corresponding to this 3 by 3 window my pixel value are z1, z2, z3, z4, z5, z6, z7, z8 and z9 that is the image, this is the image I am considering. Now, corresponding to the center pixel, the center pixel is z5 I want to determine the median value, the vector median value. So, for this what I am considering, the 3 by 3 window I am considering and I am finding the distances d1, d2, d3, all the distances I am calculating.

So how to calculate the distances? So what is the distance d1? Suppose I am considering the distance d1, so I can find a distance between this pixel and all the neighboring pixels like this, the distances I am computing, distances I am computing in terms of the RGB values. So, I am calculating all the distances corresponding to these pixels. So, all the distances I am calculating, and after this I am taking the summation of this.

So, I can write d1 is nothing but sum of the, this is the sum of the I can write the sum of the distances, sum of the distances to all pixels in terms of RGB values. So, this is a distance, the distance is the sum of the distances, all the distances I have to sum up to all the pixels because I am determining the distances to all the pixels and distances in terms of RGB value, so that is the

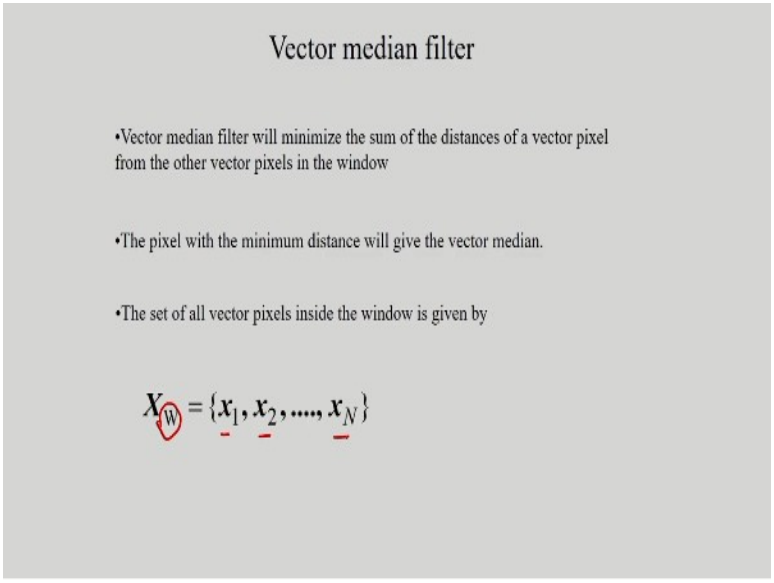
meaning of d1. Similarly, I can calculate d2, d3, I can calculate, so all the distances I can calculate.

Now, out of all the distances I have to find which one is the minimum distance. So, suppose, because already I have calculated d3, d4, d5, d6, d7 and d8 and d9, so all the distances I am calculating d1, d2, d3, d4, d5, d6, d7, d7, d9. Out of all these distances which one is the minimum distance I have to determine. Suppose, in this example, suppose the d2 is minimum, so this distance d2 is minimum. So, corresponding to d2 what is the corresponding pixel? The corresponding pixel value is z2.

So that means, the value z5 will be replaced by the pixel value z2, because the distance d2 is minimum and corresponding to this d2 what is the pixel value? The pixel value is z2, and I am determining the median value corresponding to the center pixel, the center pixel is z5. So, that means the z5 will be replaced by z2, z2 is the pixel value, the pixel value that is the RGB, RGB value I am considering, that is the vector pixel.

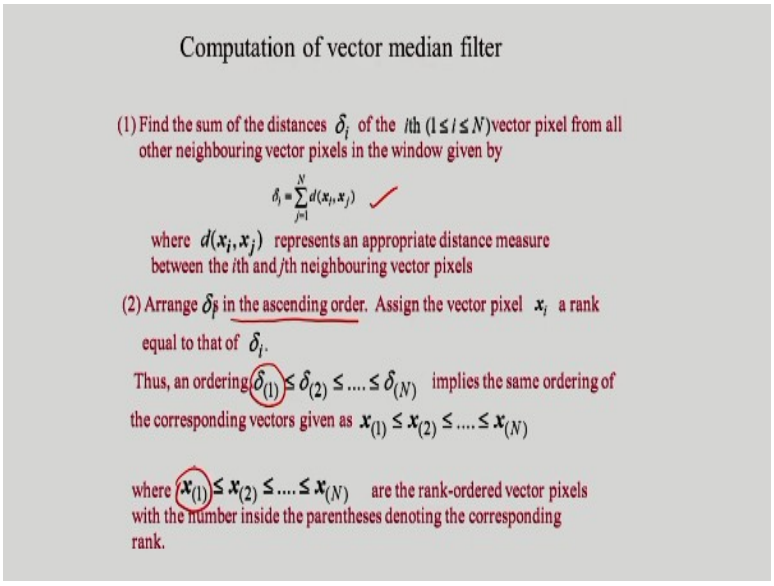
So, this is the concept of the vector median. So, by this algorithm, we can determine the vector median. So, we have to determine the distances, all the distances I have to determine d1, d2, d3. And after this is what I have to consider, I have to find a minimum distance. And from the minimum distance I can identify the pixel. And that center pixel value will be replaced by that pixels value. So, this is the algorithm for vector median filter.

(Refer Slide Time: 52:53)



So, in this case, here already I have explained the concept of the vector median filter and I am considering one window. And in this window the pixels are x1, x2, like this, these are the pixels, these are the vector pixels.

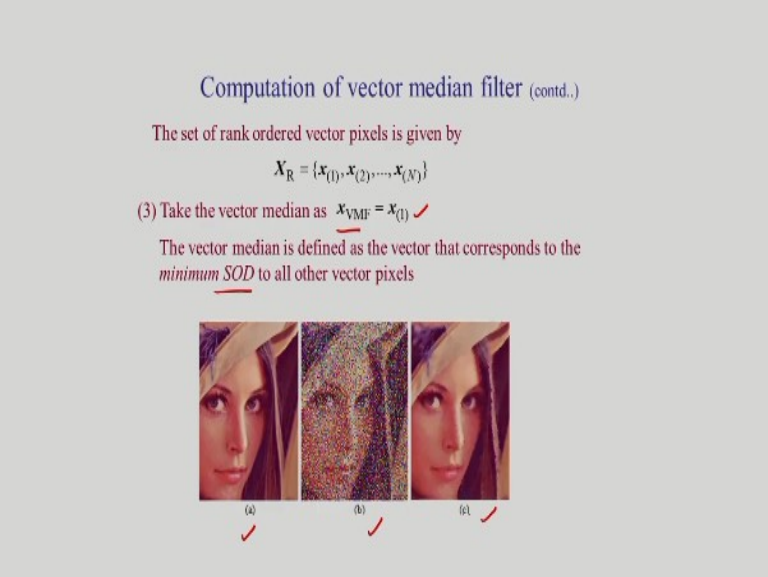
(Refer Slide Time: 53:08)



After this, the next step already I have explained that I have to find the distances, I have to find the distances and I have to arrange all the distances in the ascending order. And in this case, I

have to find the minimum distance. So, which one is the minimum distance. And corresponding to the minimum distance I have to find the vector pixel, which one is the vector pixel. Suppose, corresponding to this one the vector pixel is this one, x1.

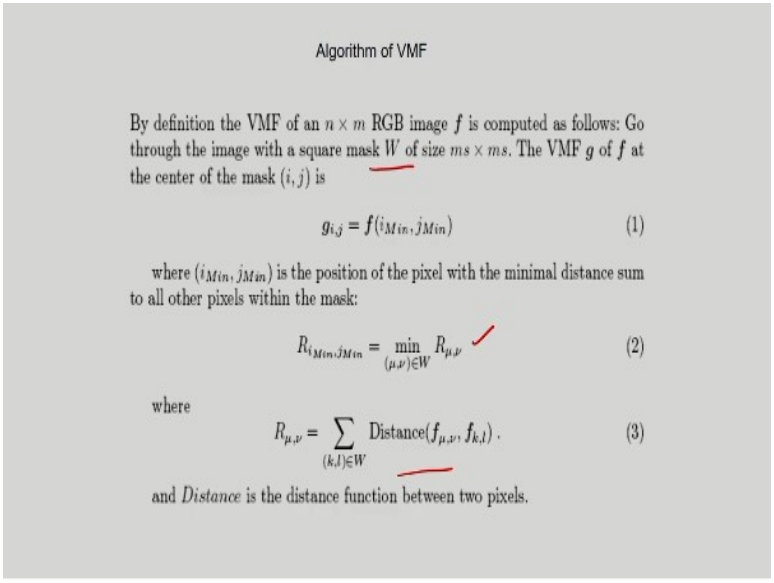
(Refer Slide Time: 53:43)



And after this what will be the vector median? The vector median will be x1, because corresponding to x1 the distance was minimum, so that is why the vector median, x vector median will be x1, that is the vector pixel. And already I have explained I am considering the Euclidean distance, sum of square distance.

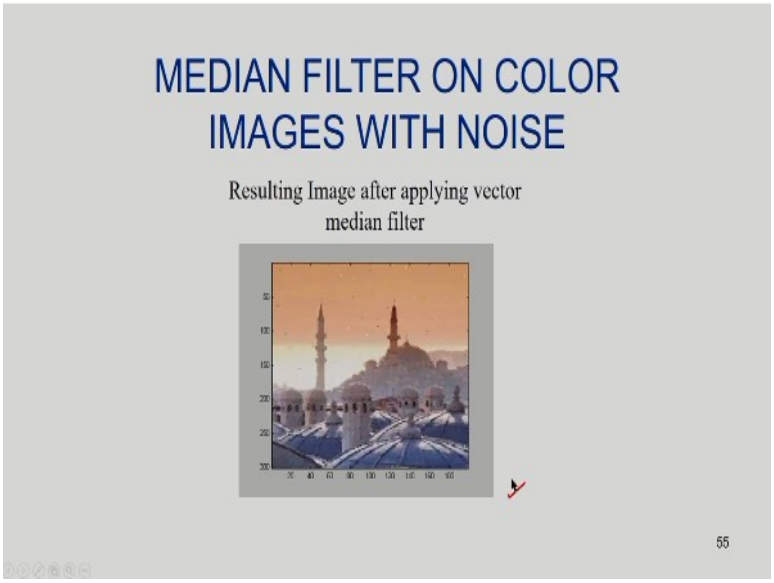
So, for this I have to consider the Euclidean distance, so already I have explained this concept, so how to determine the distance in terms of RGB values. So, here you can see in this example, this is my input image and this is the image corrupted by the impulse noises, that is the salt and pepper noise. And after vector media filter, I can get this image.

(Refer Slide Time: 54:33)



And in summary, this vector medium filter algorithm I can write like this. So, the concept is same, so mainly I have to find the minimum distance and corresponding to this minimum distance I have to find the vector median. And for this I have to consider one symmetric mask, the mask is W, that mask I am considering.

(Refer Slide Time: 54: 54)



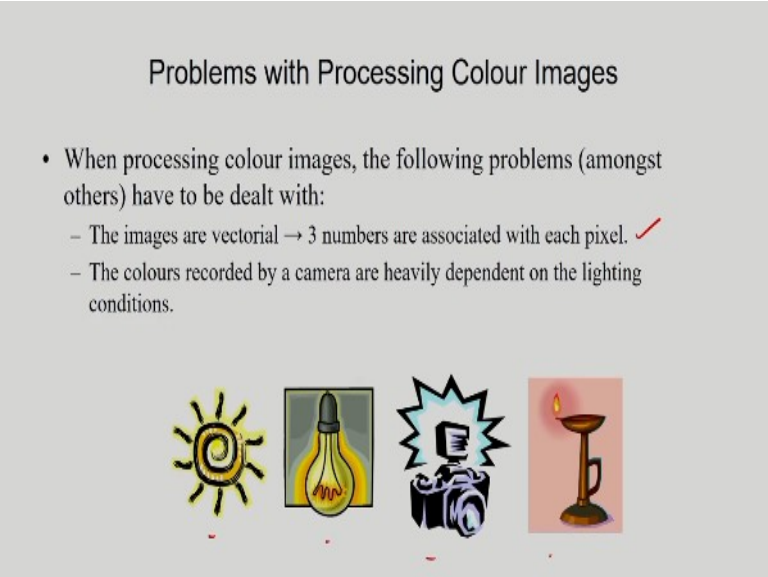
So, this is this example I am showing the salt and pepper noise, the 15 percent salt and pepper noise I am considering. And after this, the resulting image, after applying the vector median filter, so am I playing the vector median filter and corresponding to this, this is the image.

(Refer Slide Time: 55:11)

So, up till now I discussed about the concept of the vector median filter. The edge detection and the segmentation, image segmentation I will discuss later on. But for the color image also I can apply the same algorithm, the edge detection algorithm, and the image segmentation algorithm I can apply for the color image also.

In color image what I can do, I can convert the RGB into HSI and for the I, that is the intensity component, I can apply this algorithms, that is the edge detection algorithm I can apply. So, when I will discuss the concept of the edge detection and the color image segmentation or the image segmentation, then you can understand this concept. Mainly the concept is very similar to the grayscale image, for the intensity component I can apply the edge detection technique. So, this is about the edge detection and the color image segmentation.

(Refer Slide Time: 56:06)



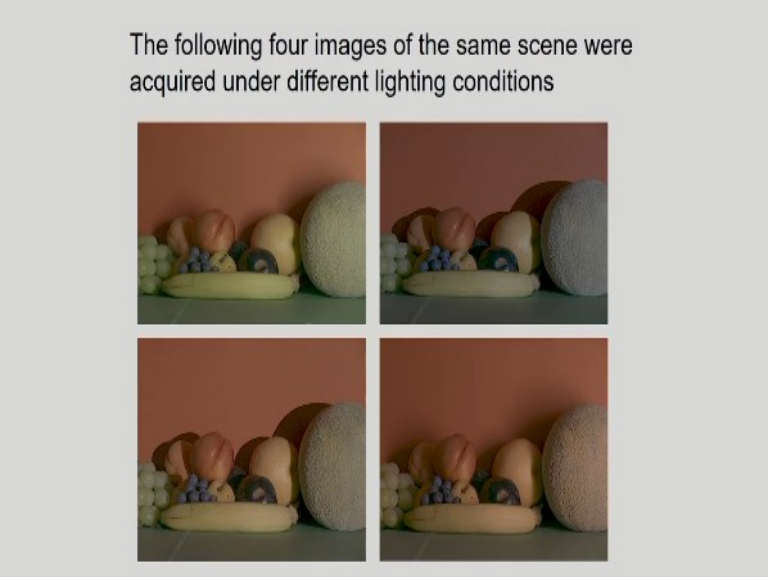
Now, I have highlighted the problems with processing of the color image. So, one problem already I have explained, that is the problem with the marginal processing. Marginal processing means the RGB components are processed separately, then the color distortions take place. And one example I have given, that is the scalar median filter. And for this I have discussed the concept of the vector median.

The another point is that color recorded by a camera are heavily dependent on the lighting conditions. So, here in this case you can see the different lighting conditions, for different lighting conditions the color will be different. So, in this case my objective is to determine the actual surface color from the image color, because in the camera I will be getting the images, but actual surface color that depends on the lighting conditions. So, actual surface color maybe different from the image color, because image color depends on the lighting conditions. So, what is the actual surface color, that I have to determine.

(Refer Slide Time: 57:16)

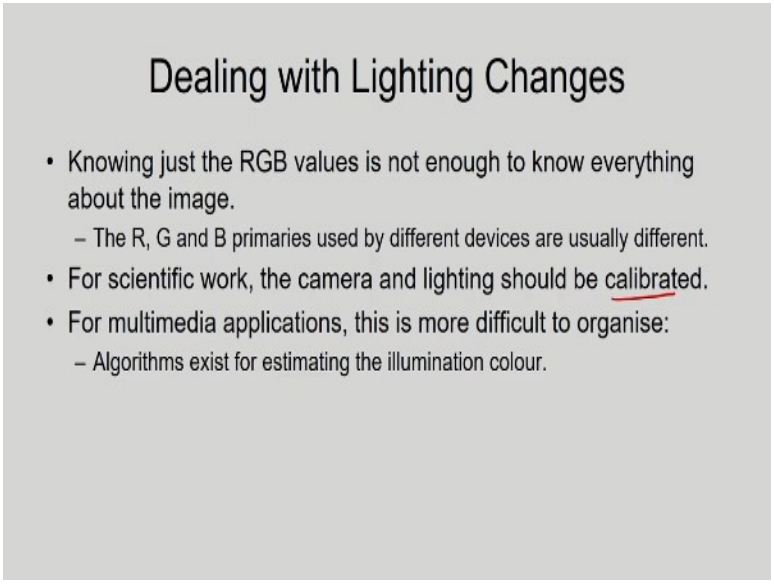
In this case you can see so, different lighting conditions and images are taken, first one is by flash, the second one is by the tungsten lamp. So, these two cases I have shown. And in this case you can see that the color is different, because different lighting conditions I am considering.

(Refer Slide Time: 57:37)



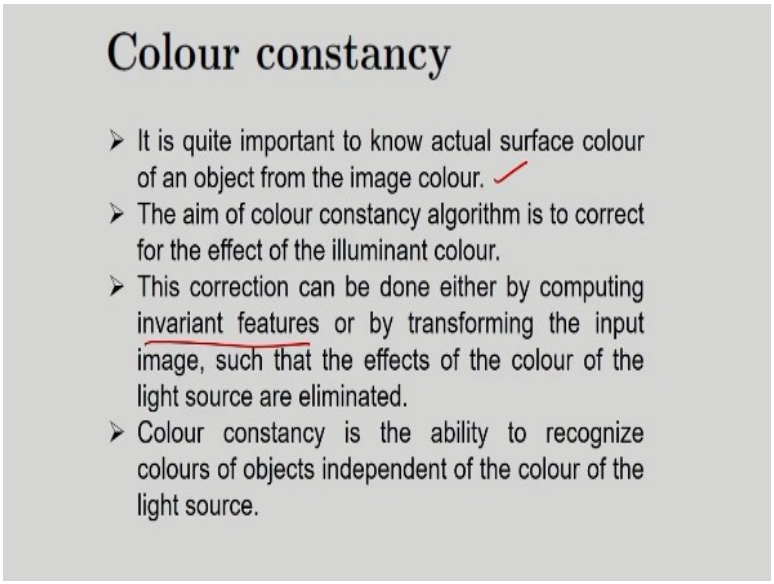
And again I am showing another effect of different lighting conditions and different images you can see.

(Refer Slide Time: 57:44)



So, that is why what we have to consider for scientific work, for scientific work the camera and the lighting should be calibrated. And for multimedia applications, this is more difficult to organize. So, for multimedia applications algorithm exist for estimating the illumination color. So, that means, my objective is basically how to detect the actual surface color from the image color that we have to consider. So, for this we have algorithms, so that we can estimate the illumination color.

(Refer Slide Time: 58:23)

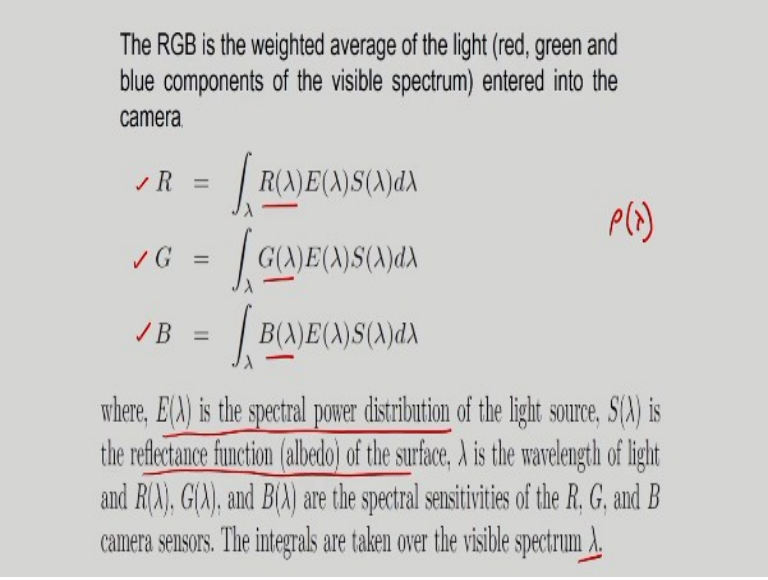


For this there is an algorithm, this algorithm is called Color Constancy Algorithm. So, what is the color constancy algorithm, then in this case, it is very important to determine the actual surface color of an object from the image color, because the image color is affected by the lighting conditions. And in this case, what is the objective of this algorithm? That is the objective is to correct the effect of the illuminant color.

Now, in this case for this what we can consider, we can determine some color invariant feature we can determine, or by transforming the input image such that the effect of the color of the light source can be eliminated. So, this is one technique, in this technique what I can consider, I can consider that color invariance features I can consider or maybe some transformation I have to consider.

So, that the effect of the color of the light source can be eliminated. So, color constancy is the ability to recognize colors of object independent of the color of the light source, that is the objective of that color constancy algorithm. So, how to determine actual surface color from the image color.

(Refer Slide Time: 59:40)



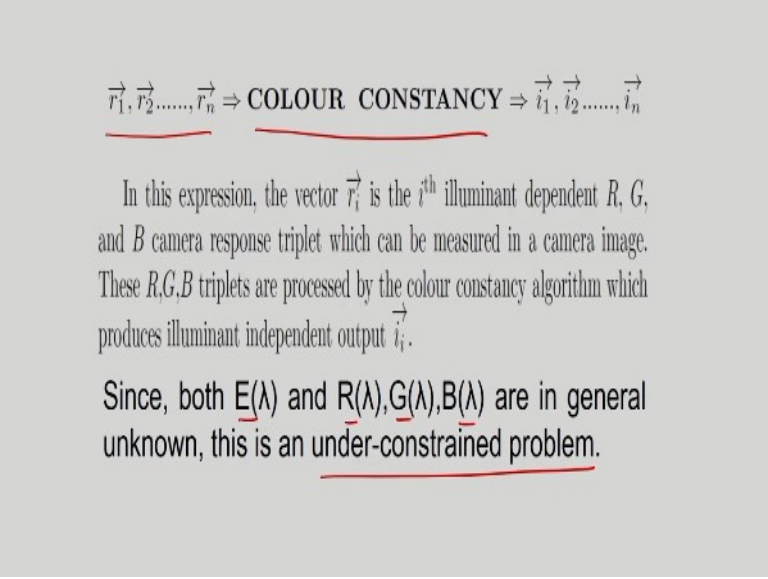
So, in this case I have shown the response of the camera, that is we have the sensors like R sensor, G sensor and the blue sensors. In a digital camera, we have the three types of sensors, one is for the red component, one is for the green component and one is for the blue component. So,

RGB response you can see here. And in this case, I am considering the E lambda, E lambda is the spectral power distribution of the light source, and S lambda is the reflectance function, that is the albedo.

Albedo, in my earlier class I discussed about the albedo, and the symbol I have shown like this, albedo is mainly rho lambda. But in this case I am showing the albedo is S lambda is the reflectance function of the surface. Lambda is the wavelength of the light and I am considering the spectral sensitivity of the RGB camera sensors.

So, for R component, R lambda, G lambda and B lambda that is the spectral sensitivity of the R component, G component and the blue component of the camera sensors, RGB sensors. And in this case I am considering the wavelength lambda, so that is why I am doing the integration over the visible spectrum, that is all the lambdas I am taking.

(Refer Slide Time: 61:03)



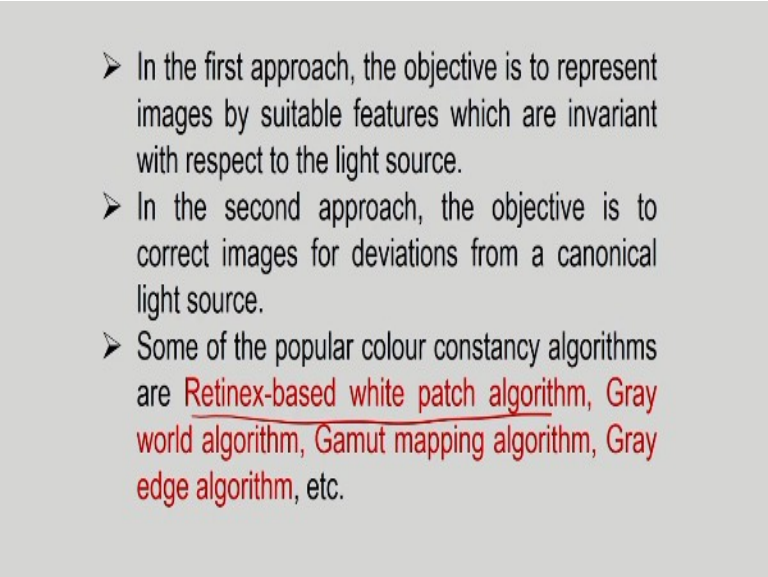
So, what is the concept of the color constancy? Here you can see I have r1, r2, r3 like this, these are the vectors. So, vector ri is the i-th illuminate dependent RGB camera response triplet which can be measured in a camera image. So, this r1, r2, all this is illuminant dependent RGB camera response.

And after processing, that is the processing means this algorithm, the color constancy algorithm I am getting i1, i2, i3 like this I am getting, that is the illuminant independent output I am getting.

That is, it is independent of the light source, that is the objective of the color constancy algorithm.

But in this case, you can see the E lambda, R lambda, G lambda, G lambda are in general unknown. So, that is why this problem is under constraint problem. So, this problem is under constraint problem because E lambda is not available, R lambda is not available, G lambda is not available, B lambda is not available, so that is why under constraint problem.

(Refer Slid Time: 62:15)



So, for this color constancy the first step process, the objective is to replace an image by suitable features which are invariant with respect to the light source. That means, I have to extract some color invariant features, that is invariant with respect to the light source. This is the first approach.

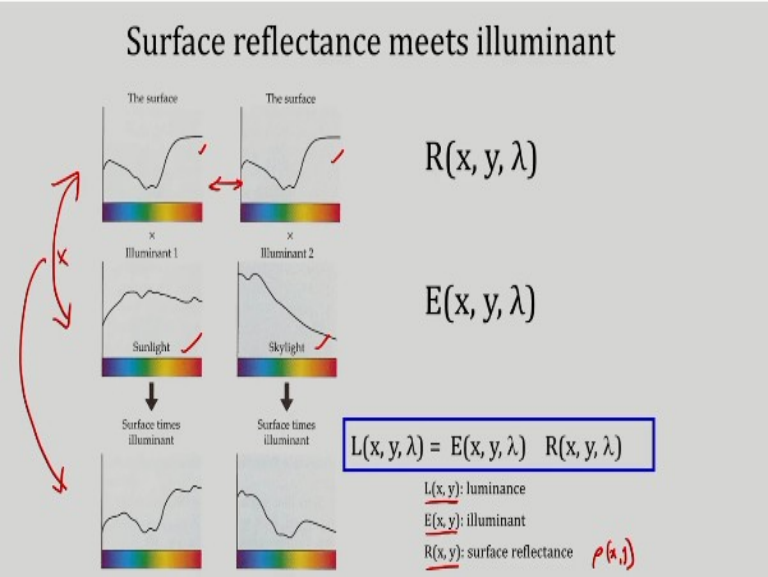
In the first approach it is not important to determine the direction of the light source or maybe the characteristics of the light source, it is not important, but in this case what is the important, the important is I have to extract suitable features which are invariant with respect to the light source.

In the second approach, the objective is to correct images for deviations from a canonical light source. So, that means, in this case the light source characteristics I have to understand and I

have to consider the canonical light source and corresponding to this I have to correct images for deviations. So, color corrections I can do corresponding to this canonical light source.

So, based on these principles, I have some algorithms, the popular algorithms, one is Retinex based patch best algorithm, Gray world algorithm that is also very popular, Gamut mapping algorithm, Gray edge algorithm. So, there are many algorithms, I am not going to discuss all these algorithms, but briefly I can explain the concept of Retinex based algorithm, otherwise, these algorithms you can read from the research papers.

(Refer Slide Time: 63:53)



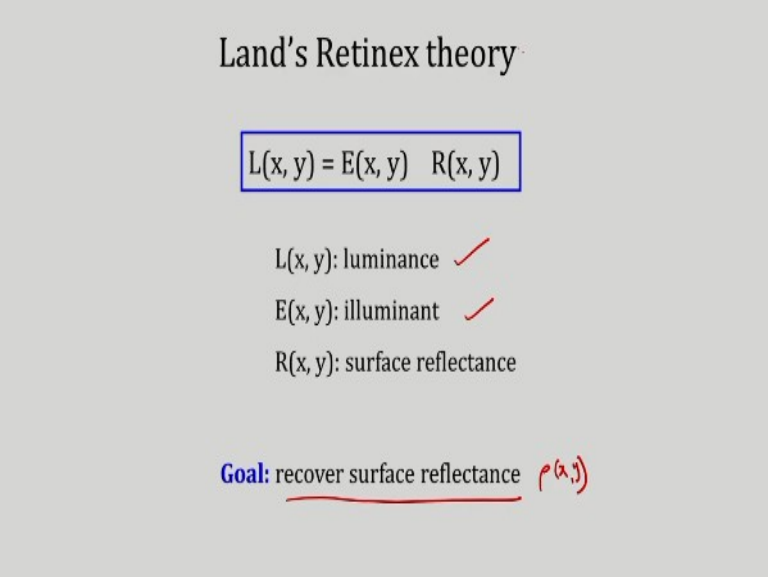
So, here I have shown the surface reflectance meets illuminant. I have shown the surface reflectance, if you see this one surface reflectance, surface reflectance I am considering that is a profile I am considering. And I am considering two light sources, one is the sunlight that is the illuminate one and skylight the illumination two.

After this if I want to determine the luminance, the luminance is the L x, y, it is the luminance and what is E x, y, E x, y is the illuminant, that is the irradiance. And R x, y I am considering that is the surface reflectance, that is nothing but the albedo, albedo of the surface rho x, y.

So that means if I multiply E and R I will be getting L, the luminance I will be getting. So in this case, if I multiply these two, if I multiply these two I will be getting this one. So, you can see the effect of the light source here, at the surface you can see the initially the surface you can see the

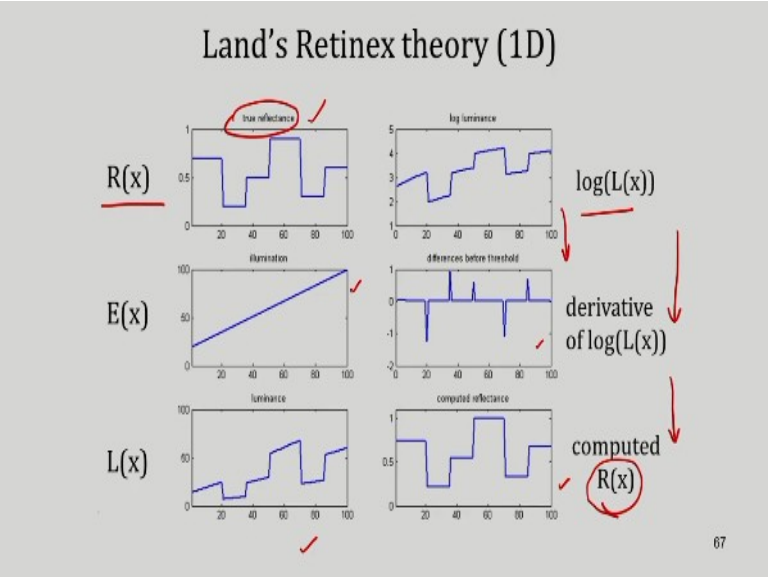
surface profile, same surface profile, but I am considering two light sources, one is the sunlight and another one is the skylight. After this you can see the effect of the light source, so that I have to compensate, the effect of the light source.

(Refer Slide Time: 65:11)



So, what is the Retinex theory? So, this L x, y is the luminance, E x y is the illuminant and R x, y is the surface reflectance, that is the albedo of the surface. So, what is the objective? The objective is to recover the surface reflectance or the albedo. So, I have to extract surface reflectance or the albedo. So, how to extract the surface reflectance?

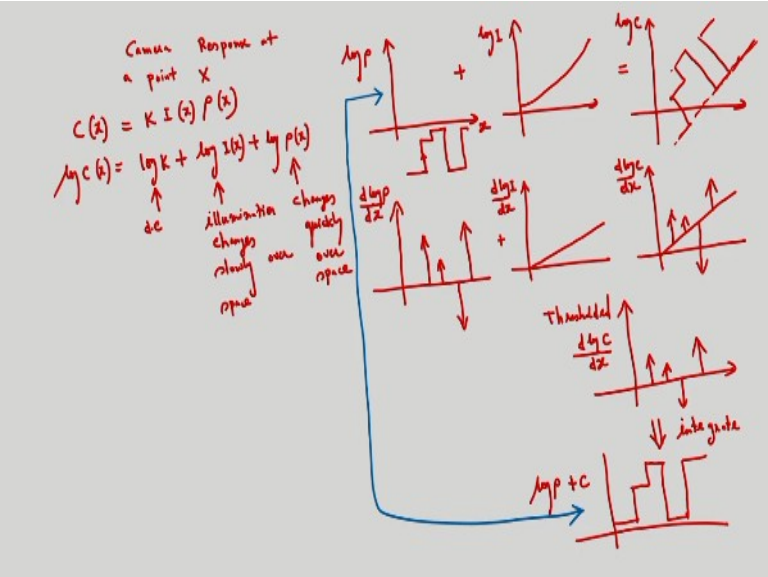
(Refer Slide Time: 65:41)



In this case I am showing one example. So, suppose the truly reflectance I am showing here, this is the profile of the true reflectance. And I am considering the illumination, the illumination is something like this the profile. And if I multiply Rx into Ex, then I will be getting Lx. So, this is Lx I am getting. Now, in this case, if I take the log of the Lx, then I will be getting this one the log of Lx.

And after this, from this if I take the derivative of log x, then I will be getting this one. And after this if I take the integration of this, then I will be getting this one. That means, I can compute the reflectance. This is my true reflectance, the reflectance I can compute it from Lx, you can see from Lx I can compute the reflectance. Here, first I am taking the log, after this I am taking the derivative and after this if I take the integration of this I can compute Rx.

(Refer Slide Time: 66:46)



So, this concept again I am going to explain, if I consider the camera response at a particular point, at a point suppose X. So what is the camera response? C x is equal to the linear camera, K is suppose the constant, so linear camera I am considering and rho x. So, if I take the log of this, the multiplication can be converted into summation. Log K plus log Ix plus log rho x.

So, in this case log K is nothing but the dc component, that we can neglect. This is the illumination, illumination changes slowly over space and this is the reflectance that is the albedo, the albedo changes quickly over space. So, illumination changes slowly over space and the albedo changes quickly over space because of varying phase angles, like in the edges the albedo changes very quickly.

Now in this case, I can show pictorially how can you recover the reflectance, because the objective is mainly to recover the reflectance. So, suppose this is my log rho, that is reflectance I am considering, this is the profile I am showing the reflectance profile with respect to some distance I am considering. And also I am considering the illumination suppose, the illumination, the log of illumination something like this log I.

So, if I combine these two, if I multiply log rho plus log I, that means if I do the addition of this, not multiplication, if I do the addition of this, then in this case, I will be getting what log c I will be getting. So, log c I will be getting, so it will be something like this, so I will be getting this

one, so this is a log c, because I have to add these two, if I add these two, then I will be getting this one.

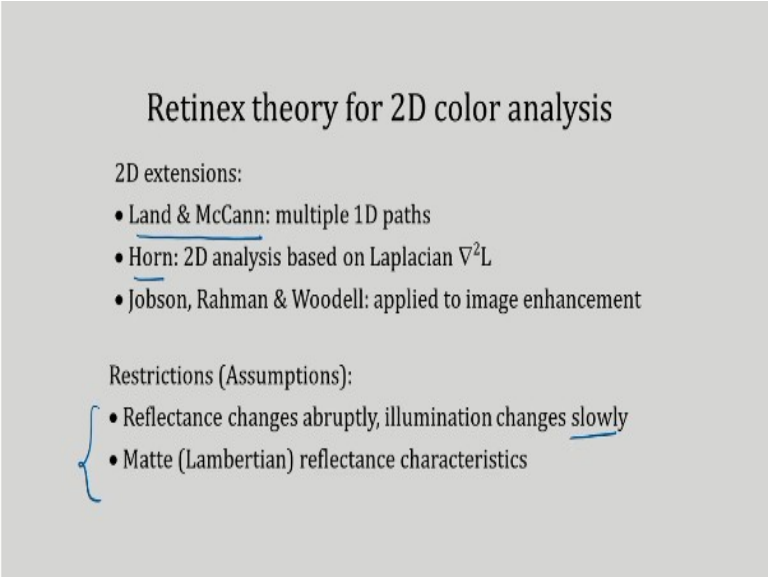
And if I take the differentiation of these, if I take the differentiation of these that is the d log rho divided by dx, so what I will be getting? Corresponding to the first jump, this jump, I have the differential. Next one is another jump is there, next one is the negative side, I have another jump, so I will be getting the this one, that is the d log rho divided by dx.

And corresponding to this if I take the differentiation of this one, d log I divided by dx, then I will be getting something like this. And in this case, if I add these two because, as per the equation I have to add these two. So, if I add these two, then what I will be getting? I will be getting this one. That means, in this case I will be getting this one is d log c divided by dx.

And from this if I do the thresholding, thresholding it if I do the thresholding d log c divided by dx, so after thresholding I will be getting these one. And after this if I do the integration of this, integrate, so I will be getting this one, that is nothing but log rho plus the integration constant is c, that is the lightness is recovered, you can see. So you can see here that is I am just recovering this one.

So, this is recovered, so that means, the lightness is recovered that means, I am recovering the reflectance that is the concept of this theory. So, I am not explaining in details, but if you want to see the solution of this problem, then you can see book or maybe some research papers you can see. So, how to solve this problem. But the main concept is this, that is how to determine the surface reflectance.

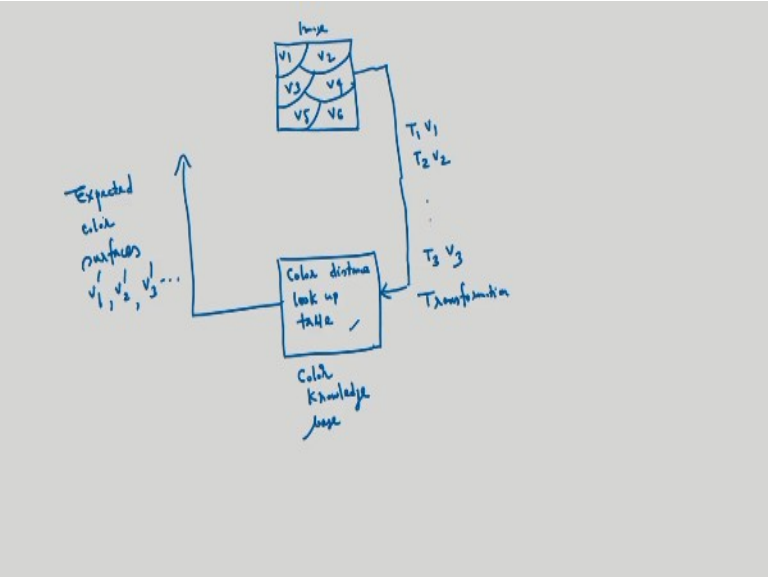
(Refer Slide Time: 72:58)



So, Retinex theory for 2D color also we can extend, like this one algorithm is by Land and McCann. So, multiple 1D path they have considered. Another one is by Horn, so 2D analysis based on Laplacian like this you will get many algorithms. But in this case we have to consider some assumptions, the assumption is the reflectance changes abruptly and illumination changes very slowly, that is already I have explained.

And also, I am considering the Lambertian reflectance characteristics. So, these are assumptions for applying the Retinex theory for 2D color analysis. And another algorithm I can explain briefly, how can you correct the color, the color is affected by the lighting source.

(Refer Slide Time: 73:54)



So, I have one image suppose, and suppose this region has some color suppose v1, this region has another color v2, this region has v3 color, v4 like this I have these different colors corresponding to this image and in this case suppose color distance lookup table I am considering, that is nothing but the color knowledge base, color knowledge base.

So, in this case I am considering some transformation, the transformation is suppose T1, V1, T2, V2, like this I am doing some transformation, T3, V3 I am considering this type of transformation, these are the transformation. And from this I can determine expected color, expected color surfaces I can determine, v1 dash, v2 dash, v3 dash like this. So, here I am going giving one example how to compensate the effect of light source, because of the light source color of the images will be different.

So, corresponding to this input image, suppose v1 is a particular color, v2 is another color, v3 is another color like this, but that is different from the actual surface color. So, I know the actual surface color that means I have to do some transformation to get the actual surface color. So, T1, T2, T3 are the transformation. So, by using this transformation I can get the actual color.

So, v1 is not the actual color because of the light source, so I am doing some transformation T1, V1, and corresponding to this transformation, I am getting the actual surface color. So, like this, I have to apply some transformation. And after this, I am making a table, the lookup table I am

making that is nothing but the color knowledge base. And based on this I can do some predictions. So what will be the expected color surfaces, I can determine.

So, suppose unknown color is coming, suppose I have one surface, so for this I have to see the color in the lookup table and I can do the prediction. So, what will be the expected color of the surface that I can determine from the color knowledge base.

So, in this class I discussed about some color models. The color models are based on the concept of the decoupling of the color information from the intensity information. So, for this the color model are HSI color models, one is that Y Cb Cr color models, one is YIQ color models. After this I discussed the concept of color balancing. So, how to do the color balancing, and that concept I have explained. After this, I discussed about the concept of pseudo coloring, so how to convert a monochromatic image into color image.

After this I discussed the concept of the vector median filter, so what are the problems of the marginal median filter and how to apply the vector median filter that concept I have explained. So this is about the today's class. Let me stop here today. Thank you.