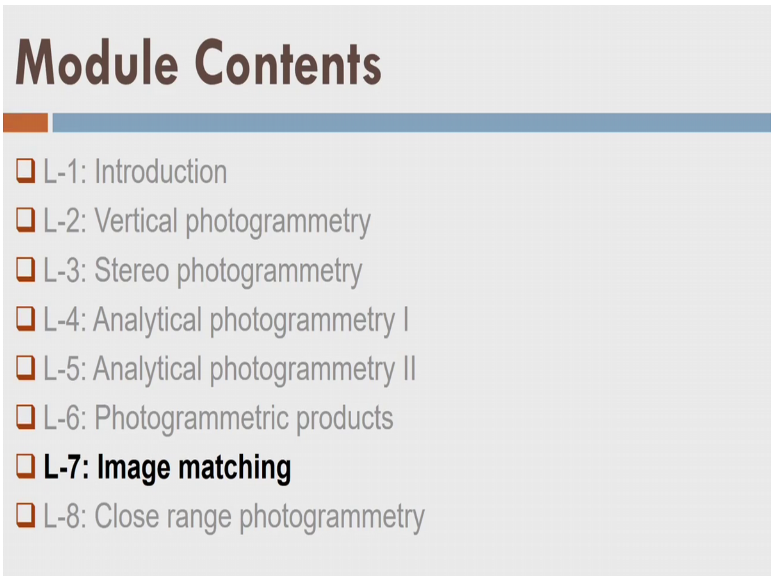
Higher Surveying
Dr. Ajay Dashora
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Module-6
Lecture – 21
Image matching

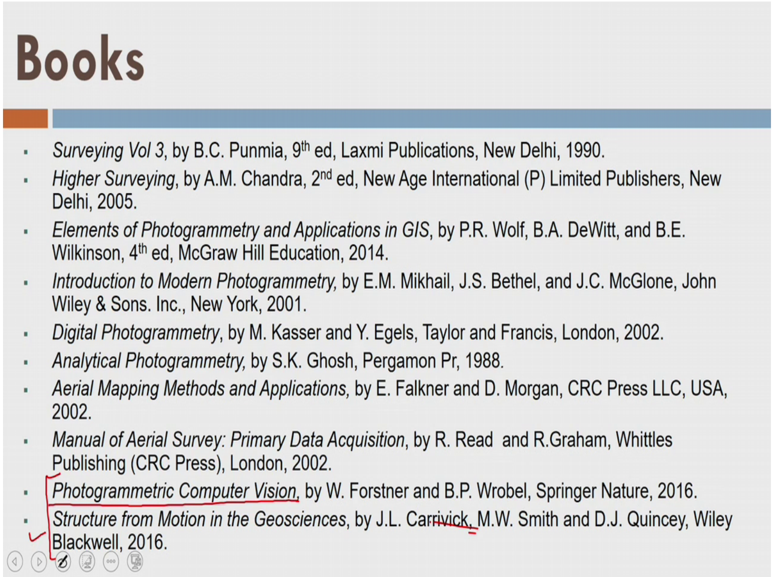
Hello everyone, welcome back on the course of Higher Surveying and we are in the module 6 on photogrammetry. Well photogrammetry has been stretching us a lot, it is the 7th lecture today, and it is more about the digital images. If you remember that in last 6 lectures we have discussed many concepts. All those concepts are equally applicable to the digital images; however, today we are going to discuss something, which is only popular or only applicable with the digital images. And that the reason we have now excluded the digital world with the earlier aspects.

That means we have been doing some work with the analogue or may be analytical photogrammetry, but now we are purely dealing with the digital photogrammetry in coming 2 lectures today's lecture and the next lecture. This lecture is the 7th lecture on photogrammetry. So, in this module we have total 8 and now we are leading to the end of this module ok.

(Refer Slide Time: 01:27)



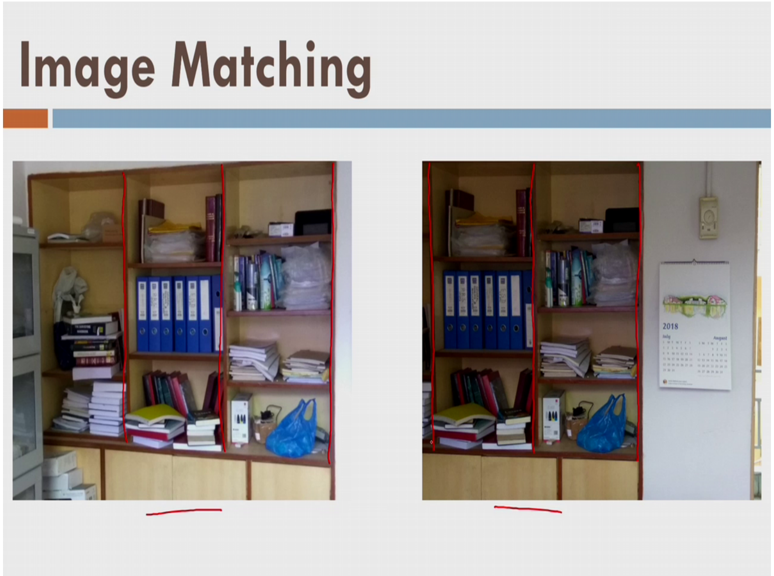
(Refer Slide Time: 01:29)



So, I would like to emphasize here about the books we have suggested 2 more books here especially last 2 books. This books are these 2 books and I would like to say that about today's lecture on image matching, there is a good introduction and good amount of material available in this book that is photogrammetry computer vision by W Forstner and so, on.

Now, what about the another book? Another book also useful for a next lecture; however, I would like to say again these 2 books are pretty expensive and hence one should join some public library or maybe in institute library hope these books are available there. Anyhow you do not need those books as such because these lectures are could enough to cover the material let us talk about the image matching.

(Refer Slide Time: 02:24)



You can see the 2 images of the same room, where there is a cabinet and in the cabinet there are lot of material is there some files, some books and lot of haphazard material has been shown there ok.

The purpose of showing these images is that if I try to match these 2 images manually like this by translation what will happen? Ideally this edge should match with this edge right you can see it very carefully. Similarly this edge should match with this edge and similarly I can say that this edge should match with this edge. Ok once this 3 edges are matched what will happen? You will get a kind of mosaic or kind of combination of 2 images where you need you find that the resulting images much wider or rather it is integration of 2 images. So, that could be a good application for image matching moreover we have many applications that is requiring image matching.

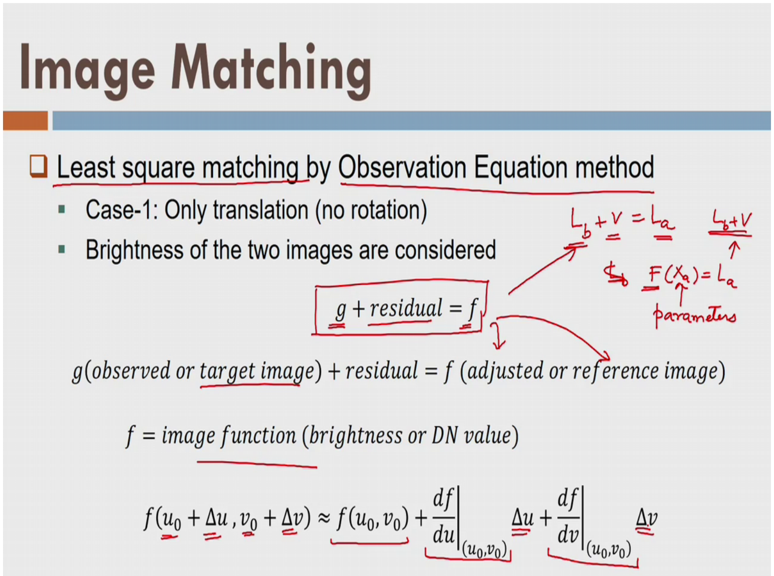
The first could be I want to do some kind of relative orientation; I also need that image matching concept there. So, that the 2 images of steopy here when they are moving like this, once they match then I can generate a 3D view and that we achieve by the relative orientation well. So, let us go ahead for the image matching and try to learn that how to do that automatically with digital computer. That is most surprising element of the image matching or the digital images.

(Refer Slide Time: 04:17)

Let us take another image here, I can that there are 2 images of the same building and this edge if you just look here it is this edge and similarly this edge of the image is this edge right and you can also find out the other common features between the two.

Now, if you again integrate these 2 images, you will get a bigger image that is combination of the 2 and that is covering a larger area fine ok.

(Refer Slide Time: 04:47)



So, how to perform the image matching that is the most important element today. And we have understand some of the concepts which are not yet popular in this course or in this module ok. The first thing is that we are going to do the least squares matching by observation equation method. You would be surprise right now we are assuming that I have 2 images and these 2 images have the same orientation; that means, exterior orientation parameters of 2 images are same, since they are acquired by the one camera.

So, I can say that their interior orientation parameters are also same. Ok as a result I need to do only the translation between the 2 images, because once they are translated like this they will be matching and then that is it. The moment it is a best match I recover or I perform the best matching or the digital computer performs the best matching will say stop here. That means, by least squares matching I will get the minimum error between the 2 images for the matching process right. So, let us say one image what I call is the observed image, I call it g and another image or I call it the reference image I call it f.

Remember in observation equation method to what did we write? L b plus v equal to L adjusted; that means, if I observed parameters are added with the residual I will get the adjusted value of the parameter. Similarly I am saying that let us say f is my adjusted or the reference image and I am trying to bring the image g; that means, if I am adding the residual, then it will become f. So, now, you can understand that how are we correlating our this concept of image matching with the observation equation method fine and then we write if you remember F X a equal to L a remember that thing where access my parameters and then we write L a equal to L b plus V fine.

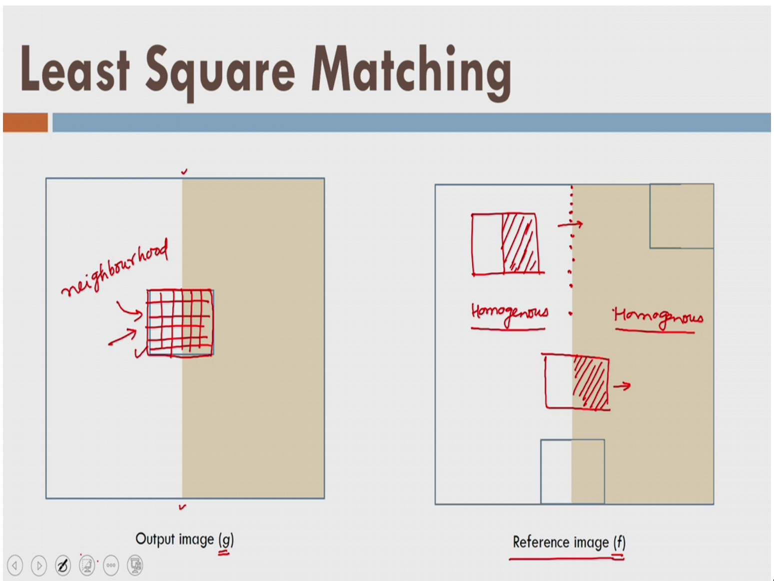
So, now we are going to use the same concepts here. So, f is my image function we call it image function similarly g is my observed image both are g and f are functions; that means, you must be surprised how image can be function ok. Let us imagine one thing that you have an image and it has some pixel values; that means, at a given x y location of a pixel I have some intensity value and that intensity value is function f. So, I can imagine that there is some surface which we call small f or may be small g, which is a continuous surface and they are representing the intensity value or the pixel value or the digital number of the image.

I am assuming that the function f is an continuous function similar function g also an continuous function in their image functions, they are not pixel numbers. So, there are x y, but they are function of x y at a given x y they have some value f and so, they are the image functions what we call here right. Sometimes you also call the target image the g image ok. Now if you remember by the observation equation method, we are first linearize our F X a remember? Same way now I am linearizing my F X a with respect to some known point call u 0 v 0. So, what is my u 0 and v 0? They are basically the translation that I want to perform between the 2 images. So, I have estimated some initial value of translation u 0 and v 0 which is v 0 in the vertical direction u 0 is in the horizontal direction in the x or y directions we can say.

And then around that point I am trying to linearize my function f by Taylor series and as a result delta u is my deviation from u 0 similarly delta v also there and so, I can write by Taylor series that this is the term I have and this here are the derivative terms into delta u and derivative term into delta v. You can easily imagine those thinks it is very easy or we are developing some concepts about image matching using least squares. So, we are doing all this thing remember as this is my image function fine and that is continuous in nature let us go ahead and try to understand what does it mean.

Ok.

(Refer Slide Time: 09:29)



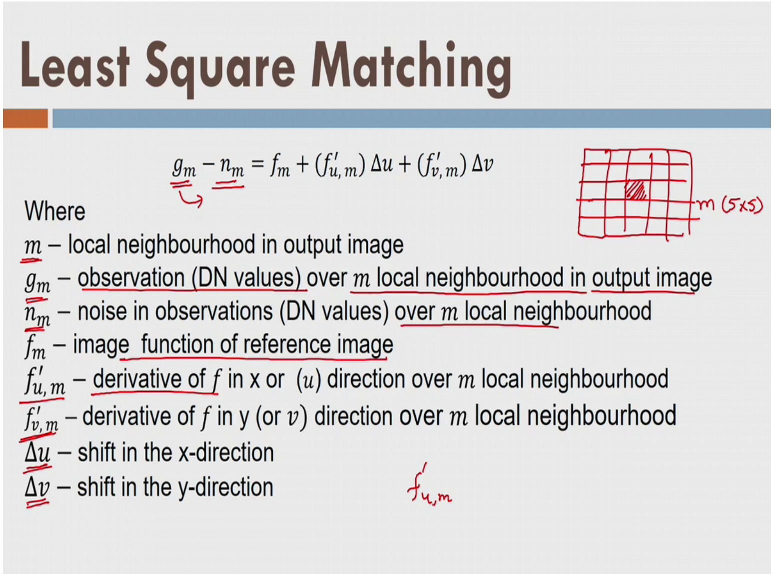
Let us see that there is an output image what we call as g and there is a reference image, if I put the animation like this is my f here and I want to match these 2 images; that means, I am putting the same image and I am trying to match 2 images. Ideally if they are matched then they will be having the same edge here and it will match exactly. So, now, just see that this is my window and I am trying to move on this way right. So, if I take this window shown in the animation and for try to move on this way right, so, if I take this window as I shown in the animation and if I try to move over that what will happen? So, it may not detect that what is the change in the image matching process, but what if I move this same box on the x direction let us see like that like this now I am moving it.

So, definitely when it passes through this edge this box will find out lot of changes remember because of the change in the f function that is intensity values, because here there is a homogeneous intensity white colour and there is another homogeneous colour that is only single colour I have. However, at this edge at this point you can understand that there is a change in the f function the image function.

So even if I take this particular piece here from the my g image, and I try to put and try to slide them minimum variation will be in the homogeneous areas, but there will be maximum variation or I can say you will find out that if I place here this kind of thing here. So, there will be some variation you will find out; however, the moment the it is matched like this while moving in the x direction here you will see that exactly it will matching and as a result, I will have minimum variation between the 2 and I will say that yes I have done I have achieve the least squares matching of 2 images.

So, I hope you can got the concept that I am taking some kind of box here, that I call it neighbourhood. And I am using this neighbour in order to detect where this neighbour has the same intensity value or the minimum variation in the target or output image or the input image whatever in image f. So, I am taking this box as my neighbourhood; that means, I am saying that whatever number of pixels are there like this; I am taking this neighbourhood m by m and I am trying to bring to this thing there on the my reference image and trying to see where this variation will be minimum. Now you may find out over this process when I am moving the things on x pixel by pixel, you find out this point this point this points these are the points where 2 images should match on this edge and this is we are trying to achieve in this process ok.

(Refer Slide Time: 12:53)

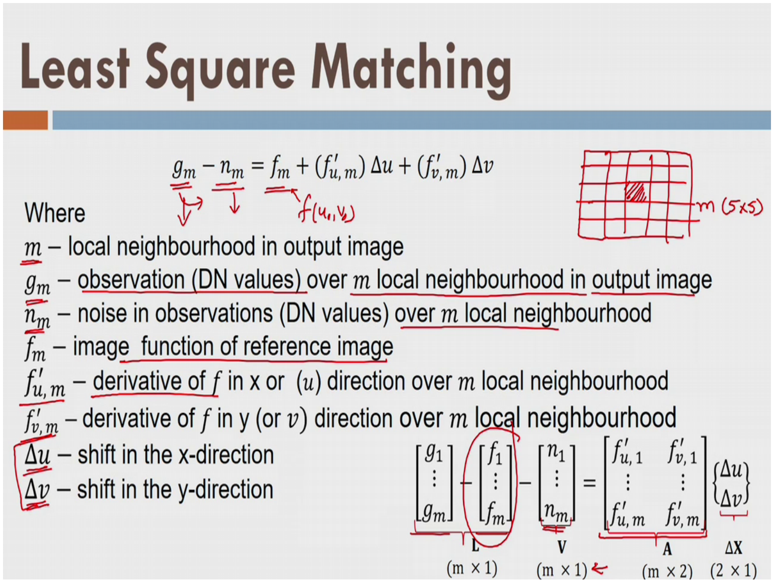


So, let us say my residual is my noise, noise means some kind of disturbance in the signal or some kind of I can say confusion or some kind of you know deviation from the ideal image. So, g is my image observed image and again what is my m? M is my local neighbourhood remember the size of the box, it is my m; deciding the let us say there pixels like that, I am trying here 1, 2, 3, 4, 5. So, it is regarding this pixel it is called 5 by 5 neighbourhood because there are 5 rows of pixels and 5 columns of pixels generally we take odd number 3 by 3 7 by 7 and larger the this neighbourhood, I will have more confidence in my work because the larger neighbourhood is confirming that what it is detecting in the reference image large neighbourhood will definitely take more time for the calculation.

And that is some kind of advantage as well as limitation the larger the neighbourhood I will take lot of time, but I will be more confident in my calculations. On the other hand smaller the neighbourhood I will take less time, but I will be less confident about my calculations right. So, g m is nothing, but observation that is the observed image over m local neighbourhood in output image right ok. N m is my noise again over the m local neighbourhood and it is applicable to image g here ok; f m is image function of reference image or what we call as the ideal image or I want to achieve that; that means, I am putting the g over the f image such that my reference image is not going move for rather g is going to adjust itself; that means, my data is g that is output and its going to adjust with the reference ok.

So, what about the derivative of f? So, this is my derivative in x direction I wrote it its very specifically because u is the translation in x direction. So, I write it f dash u and again over the m neighbourhood, I am calculating my derivative over m neighbourhood and will see what is the meaning of that in coming slides. Similarly in v direction or in y direction f dash is my derivative, and delta u and delta v are my shift; that means, with respect to u 0 v 0 do I need to shift in positive x or may be negative x may be positive y negative y whatever the values will come here and we will update our u 0 v 0, and we will try to shift till we get the minimum values of the of the function in the least square sense.

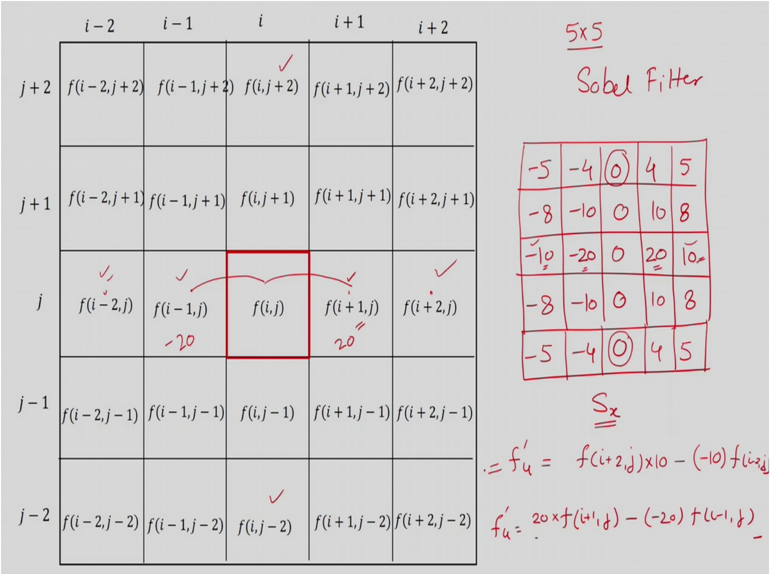
(Refer Slide Time: 15:49)



So, let us go ahead. So, now, this is my function written here; remember I have written these for m number of neighbourhood this also for m number of neighbourhood like this, like this and I will talk about what is this value? Since it is the if you remember f m that is why we say that we have linearize this thing. So, the value of the function call here f u 0, v 0 this is this here and this is also here this one is nothing, but f u 0, v 0 right. This is my noise here over n neighbourhood and these are the derivatives now the important aspect is how to calculate the derivatives for the f or maybe for the g ok? You see here what could be the size of these matrices. So, I am writing m by 1 fine.

This is m by one again that is my a matrix and it is my delta x matrix right there are 2 unknowns. So, we have this thing v matrix and now I have written in the form L equal to v is equal to a x minus l form and I can find out the solution here my least square sense ok

(Refer Slide Time: 17:03)



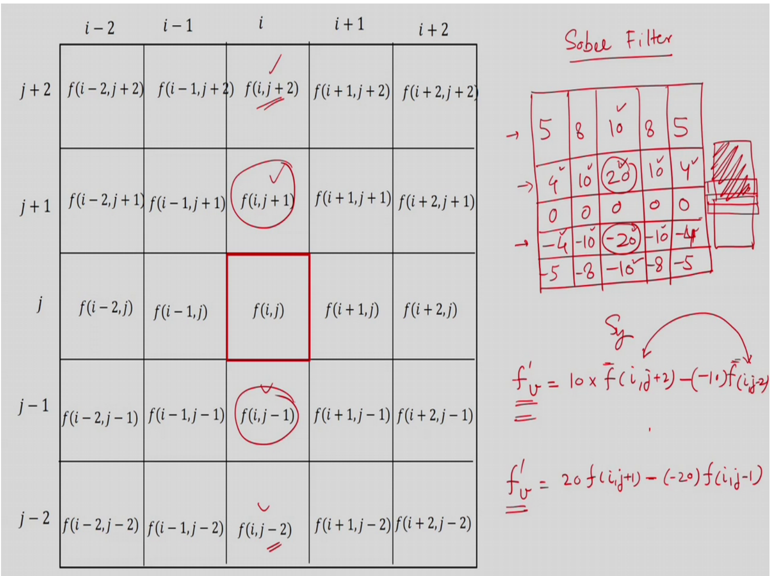
Let us say there is some neighbourhood m by n, here I can say there are 5 by 5 boxes are there, fine you can observe it and what we called here it is called Sobel filter right and what does it mean? This filter has some values like this, I am specifying like this these are all the values here they are some specific values. So, let me first fill this thing here like this now let us see this is my f image? F is my image function as we already discussed we defined ok.

So, let us see that I try to fill these values like this. Now let me fill some value here minus 5 plus 5 minus 4 0 plus 4. So, let me this side 0 0 0 0 and then we have some values like minus 8 plus 8 minus 10 plus 10 minus 10 minus 20 plus 20 plus 10 and then we repeat these values ok. So, this is my Sobel filter S x because it will detect the edges in x direction, and then you write that let us say this values stands here right and this value stands here fine.

Similarly, I will take this value and this value since it is my S x the Sobel filter in x directions I am taking these 2 values, this value and this value and I can write my f dash let us say u equal I use this value also and this value also. So, what does is mean? So, f i plus 2 j into 10 this value minus minus this value into f of i minus 2 j and that is what we call finding out derivative one more way let us look into this thing possible now I am using these 2. So, how to write it let us say 20 this value is there. So, I multiplied 20 here and minus 20 here.

And I will take the difference of the 2 and do the calculation. So, it will be 20 into f of i plus 1 j minus minus 20 f of i minus 1 j. So, this is the way we calculate our derivatives we calculate f dash v and for that we need to have a different Sobel filter what we call S y. The filter that detects the edges in y direction ok. So, I hope you [get/got] got the idea how to calculate the derivative using image and using this Sobel filter.

(Refer Slide Time: 20:55)



Now, let us once again look into the Sobel filter and this time we are defining the Sobel filter for detecting the edges in y direction let us do it again. So, let us see all these are values for f function right.

Now, let us define a Sobel filter again and this time the Sobel filter is slightly different or other it detects the edges in y direction and how to detect that will see. Let us say 4-5 lines 2 3 4 and 5. So, the s y Sobel filter is given as 5 8 10; 8 5 then ill repeat this line here with minus values similarly this line here with minus values right. So, this is why my Sobel filter to detect the edges in y direction why because

If there is an edge here for example, let us say there is like this that mean there is a red area and then this is a white area, and if I put this 0 line over there like this what will happen then I will take the values over there I will multiply with these values a white areas and I will multiply the values with these values here ok.

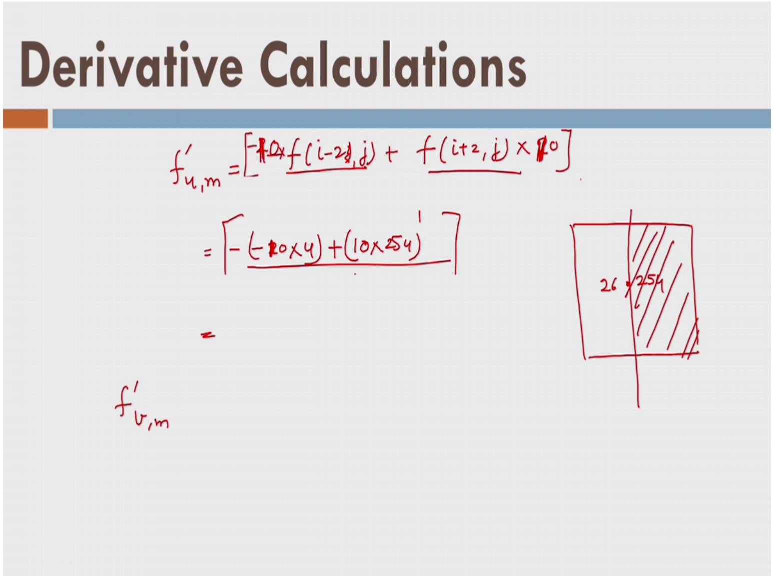
Then I will take a difference and if there is edge it will be enhancing of the image or enhancing of the edge. So, how to calculate derivative? Now f dash v equals to I will take this or this or maybe this and this let us see how to do that. So, I will take this and this. So, let us say 10 into f of this value i j plus 2 minus minus 10 into f of this value that is i j minus 2. Similarly I can calculate my f dash v also in different way or different value using this value here and this value here in the Sobel filter and this value in the image and this value in the reference image as 20 into f of i j plus 1 minus minus of f of i j minus 1 I hope you got the concept how to calculate f dash v right.

Now, the idea here is let us imagine you have some kind of image, where as I explained you here if you are in a homogeneous area; that means, colour is not changing right there is only one colour imagine that there is a black and white image where half of the image is black and half of the image is white. Suppose you are putting your Sobel filter in black area or maybe white area. So, what will happen? My all the values of the image f are same and as a result even if I multiply with this logic, I will get my derivative value as 0 you can try yourself. Why because these values are same fine and as a result I am not going to get any advantage there, but if there is a some value at the edge; that means, there is white area here and black area here what will happen now if I take the difference of these two?

So, this value will be very very high you see this value in case of black it will be 0 and this value if if it is in white are, it will be less at 255 or 250 or something like that very high value. And as a result this difference will be very very high automatically because there is an edge. So, now, you can imagine that in presence of edge this derivative value will be very very high, but in presence of no edge these derivative values are very very minimum under side.

Now, I have already explained you how to calculate the derivative.

(Refer Slide Time: 25:33)

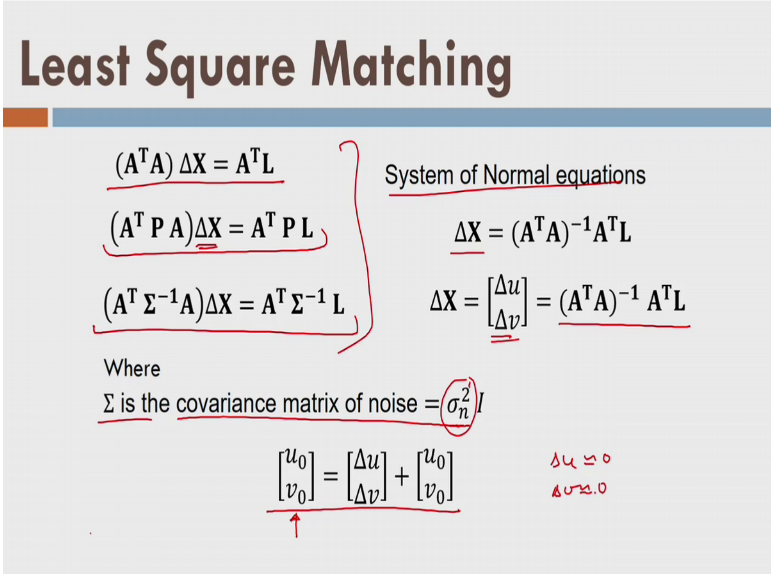


So, let us say we have calculated the derivative f dash u for n neighbourhood and equal to given by for example, f of I can say here if you remember i minus 2 j and then minus plus f of i plus 2 j and this is multiplied with some factor let us say positive 20 I do not remember now, the values we can find out from the operator and here I am multiplying here with minus 20 here like this. These values are same. So, now, what will happen? I hope these values were minus 10 and minus 12 as I remember now no problem does not matter these values are same.

So, what will happen if there is an edge in the f image what will happen? This value into 10 minus this value which is on the edge like this, at this point this is one homogeneous area and this is another homogeneous area. So, let us say the value is here 254 and here value is 26. So, then I will calculate my value as in to minus 10 here into plus 4 into minus here plus 10 into 254.

Now, you can imagine what value you were getting. So, what about the value is very high value? So, it is indicating very high number and that is the way we understand that how the moment there is edge is there, it will be very high value of derivative and this is the way we define the derivative same way I can find out what is my f dash v for m neighbourhood as we have explained in the last slide.

(Refer Slide Time: 27:23)



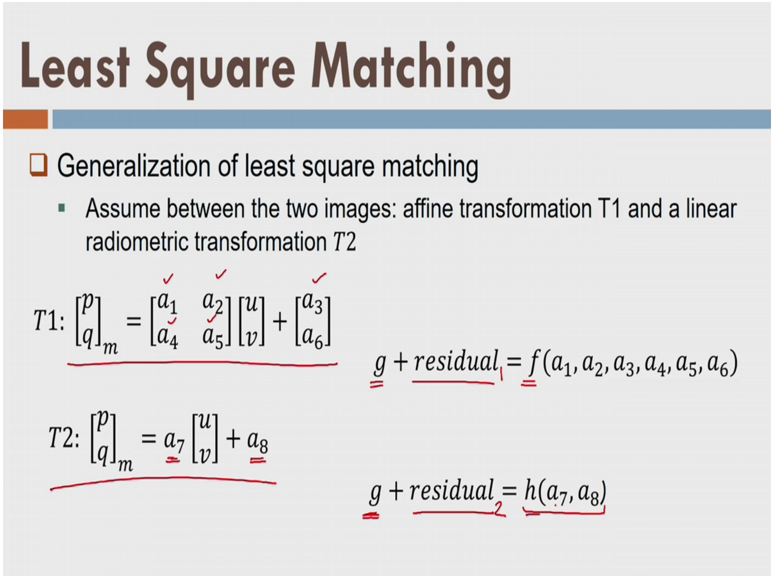
So, let us see that we have calculated f values and everything for m m neighbourhood, and then again now I will use my standard least square solution where delta X my delta u delta v and I can find out this or by this right.

These are the standard equations that we have learnt already in the lecture of observation equation method in module 4, you can view them again if you need right. To sigma is my covariance matrix of the noise remember noise is the residual error and I can write it sigma n square that is the reference variance if you remember sigma n square into i identity. Because we know that we assume here that noise is more or less same on each pixel right as a result I can write sigma n square and so. So this is my system of normal equation these 3 equations ok. So, I can write delta x like this further my delta x nothing, but delta u delta v and then I will find out this way.

So, I will find out my first value of delta u and delta v; that means, from u 0 v 0 how much I need to further shift. So, I need to update my u 0 and v 0 like this. So, the moment I updated that what will happen? I will use the new values of u 0 and I will try the same process again repeat the same process again for each pixel in the reference image as well as in the g image that is output image. Now you will be surprised a little surprise will should be there that what is the purpose of this repetition. Remember at certain stage my delta u delta v ultimately will become minimum or very close to 0 that means, my delta u become 0 and delta v become 0 because that is giving you the minimum value of delta v and delta.

And as a result we say that now we have match our 2 images and at the age or not we can detect it personally by viewing it. So, that is the process called least square image matching or least square matching for image matching.

(Refer Slide Time: 29:29)



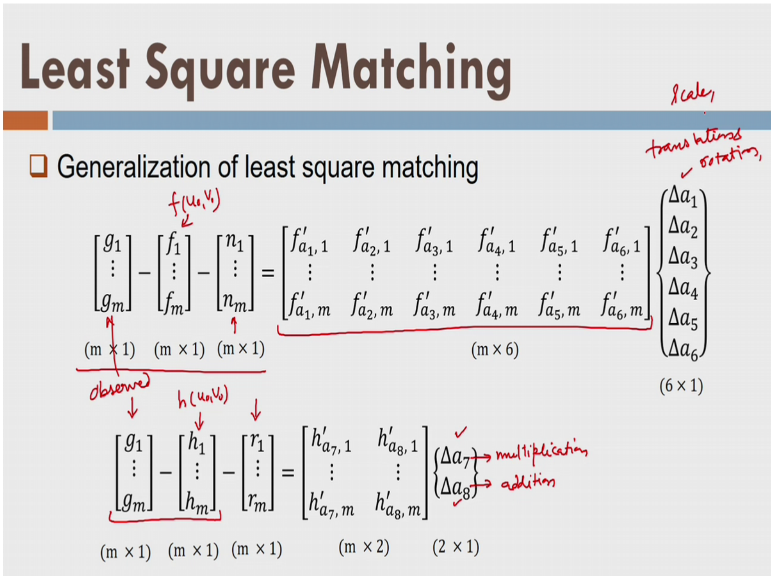
Now, let us sink that we have 2 images and in the last case for the purpose of explanation I have taken very simple example of 2 images where there is only translation in x or y direction was there. However, now imagine that the e o parameter and i o parameter of the images are same, but I am trying to match one image by this way. So, definitely if there is edge which is vertical, it will not match like this I need to rotate also my image I need to translate also my image like this and this all the motions.

So, what we do we do the affine transformation and the equation is like that, we have all we learnt the affine transformation where these are my translations and these are my rotation matrix elements. So, it is also called six parametric transformation. Moreover I am also adding something here, and that is nothing, but a linear transformation; that means, I am saying my g if I add some residual, then it has some function h and it is kind of linear transformation; that means, my image function is also changing fine and now putting it little more thing here. Not only the transformation or the x and y and theta and angle no I am saying that my image function is also changing right.

So, I am writing by this way. So, these are my nothing, but a 7 is my multiplication factor at a 8 is my translation factor; that means, I am adding some value a 8 to my image function, and I am multiplying my image function with a 7 and I am getting another image function I am saying that now what about the residuals? These residual it is one and it is 2 because they are different remember that one I add residual one to g I will get f; similarly if I add residual 2 to my g I will get edge right.

Now, let us make our life little complicated, but little better ok.

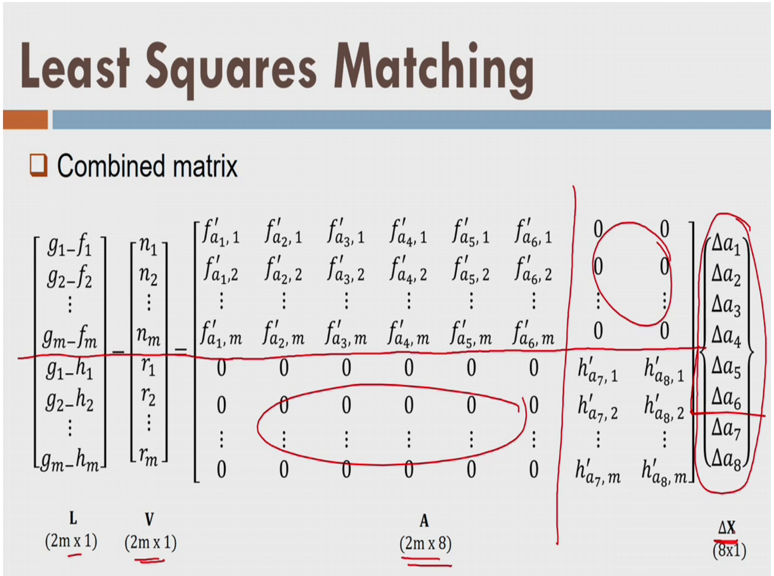
(Refer Slide Time: 31:29)



So, again I am writing the same function here where this is my f u 0 v 0 right, this is my noise. So, this is my derivative for these parameters of f you can find out analytically, and then you can put the pixel values there again using the Sobel operator you can find out ok. What about this one similar fashion I wrote like this, my u 0 v 0 and these are these values. G is my observed values like this over n neighbourhood they are same. So, they are basically repeating I can say here.

However these are not repeating. Now I am writing my noise here remember I said they should be a different noise. So, I am writing my noise by r here over n neighbourhood. So, my neighbourhood is say m it would be 5 by 5, 3 by 3, 7 by 7 and 9 by 9 and so, on right. Now here I want to calculate these parameters that is one is a multiplication factor in my image values, and here it is addition value. So, it is just kind of y is equal to m x plus c. So, my function image function is also changing right with respect to the I can say by linear way right and here they are my translation and rotation factors or rather I can say translation rotation as well as a scale fine. So, 3 factors are there and there are total 8 factors we have ok.

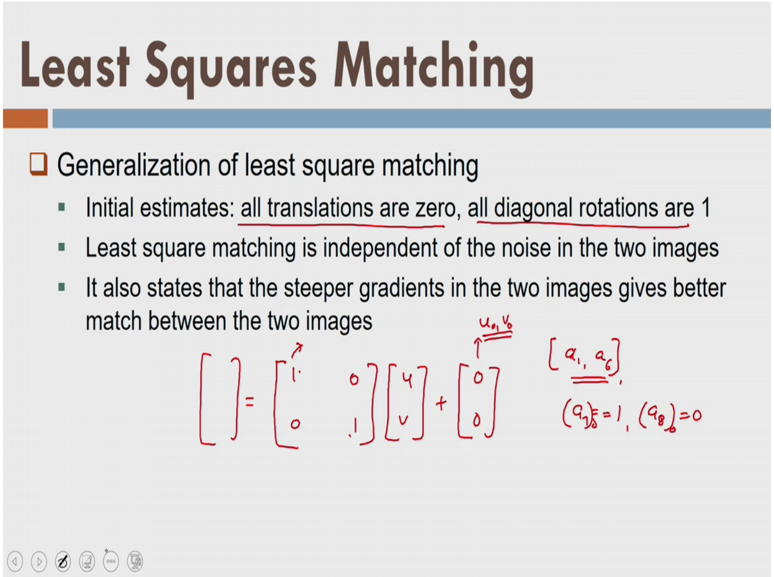
(Refer Slide Time: 33:17)



Now, let us integrate the 2 matrices that is here you can see that I can construct this kind of metric system when like this is like this year. So, that you will see you can see that I can construct this kind of matrix system well like this it is like this here. So, I have added you can see here I can partition this matrices here because it is complete 0 this is complete 0 and if you are writing a software for you, you need not to store all these values because 0 is anyhow is not going to give you any value for that purpose of calculation.

And as a result we always keep these 2 matrices separate and we do the matrix partitioning and we try to estimate these unknowns together using matrix partitioning right. So, now, you can see all these things are very very easy and again I can write the same equation here that v is equal x minus l v l a and delta x here, again I will use the same logic to calculate this thing.

(Refer Slide Time: 34:13)

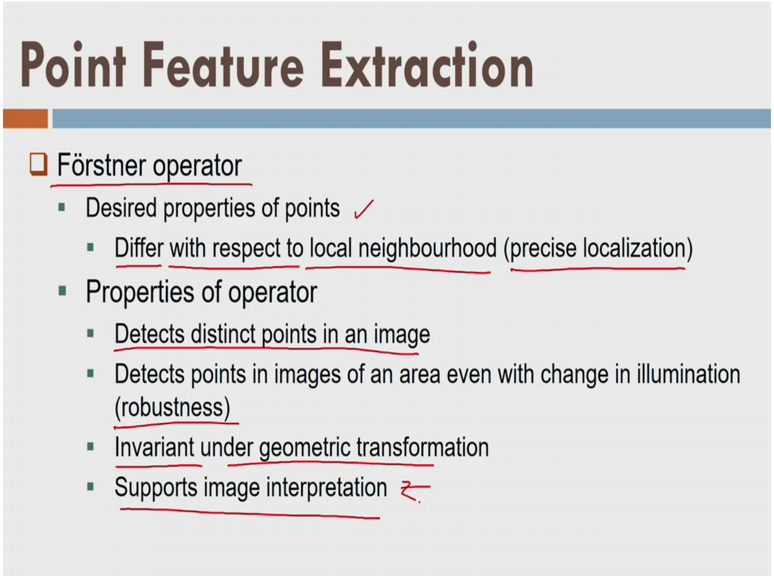


So, what about the initial estimate of u 0 v 0? So, I can say all translations are 0 and all diagonal rotations are 1.

So, what is the meaning here that my a 1 is 1 0 0 1 this matrix plus u v plus what about the the u 0 v 0? I will say 0 and 0 translations they are my u 0 v 0 and I get some value here. So, these are my initial values of a 1 to a 6 similarly I can also find out a 7 and a 8 again a 7 should be initial value of a 7 is my 1 and a 8 is my 0 initial values.

They are all initial values we are talking about right. So, here u 0 v 0 that they are my translation factors one among the six here, and these are my scale one or I can say rotation one. So, these are 1 1 right. So, I can now understand that if there is a steeper gradient; that means, if there is a clear edge. So, what will happen? One area high value and another area very low value, so, there is a edge. So, I will enhance that edge that my that is the idea here. So, the higher this difference easier for detection right. So, this is what we call the least squares logic for the image matching and what sometimes we call least squares matching for images ok.

(Refer Slide Time: 35:55)



What about the point feature extraction ok? I have given you one situation before that I want to match 2 images; however, now I am giving you a situation where I am seeing clearly that I want to detect the point what is the meaning here? Meaning here is very simple I am changing the context now. Let us say there is an image and if there is a distinct point; that means, let us say some ground control point, that is visible in my image and that control point is also visible in the ground surface, but I want to force detect my ground control points on the images before collecting the data in the field. And for that purpose I need to detect them can I do it digitally can I do it automatically let us see this thing. So, this thing is that that the popular point it should differ with the local neighbourhood here.

That means it is called the precise localisation, I can localise that point very easily compared to its surrounding that property should be there. Further I should say that it should detect my points that is I am basically using the Forstner operator and it has some characteristics, this is the kind of requirement it has the moment we fulfil this requirement, it can detect distinct points in the image and it will detect the point in images on an area even with change in illumination that is the robustness; that means, even if the illumination of the scene or the image is changing, it still it can detect the very very pinpointed point which are comparatively different from the surroundings ok.

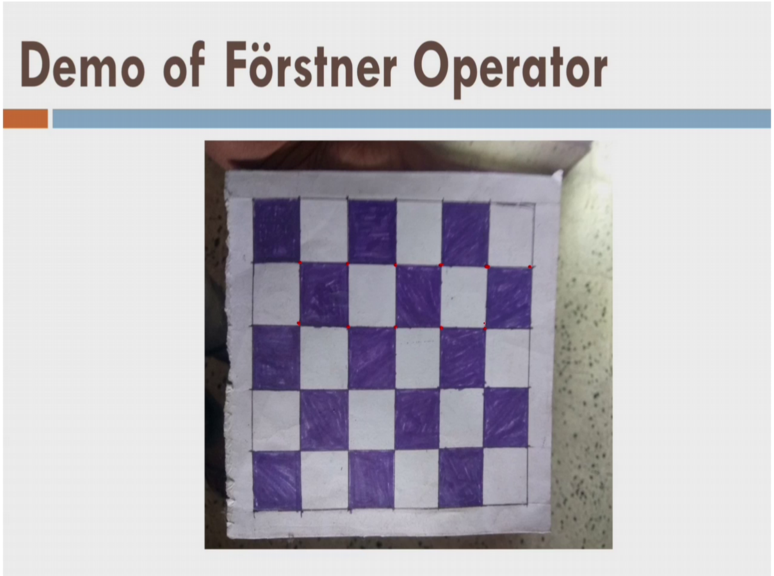
It is invariant under geometric transformation fine and then it will it supports the image interpretation. That means, once you detect the control points or the some points which are distinctly with their surroundings, then you can find out what are these points and what are these features indicating. So, that is called that we call as image interpretation. So, my Forstner operator does it for me.

(Refer Slide Time: 38:00)

Let us see what is the Forstner operator again I am using one image here and I want to show that I can say that this point is my distinct point. Similarly I can say that let us say my this point it is distinct because it is completely different from surrounding.

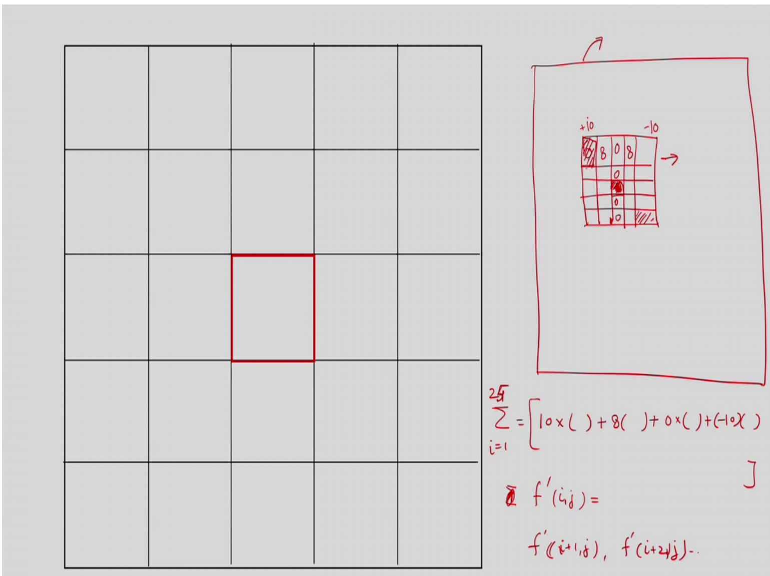
So, many points I can mark for example, this point on the file it is distinct from the surroundings and it is a kind of one of the control point if my terrene is looks like that. Similarly I can say this point also I can say this point also any point that is different from the surroundings is my distinct point fine right. And we can mark many points on this image and even this point is also a distinct point, because here it is yellow, here it is white, here it is something else, here it is something else all the surroundings if I look at. So, they are good points to detect.

(Refer Slide Time: 38:55)



Now, let us see this particular example where I can say this is the points which are distinct points and I want to detect these points fine using Forstner operator right all these points I want to detect.

(Refer Slide Time: 39:17)



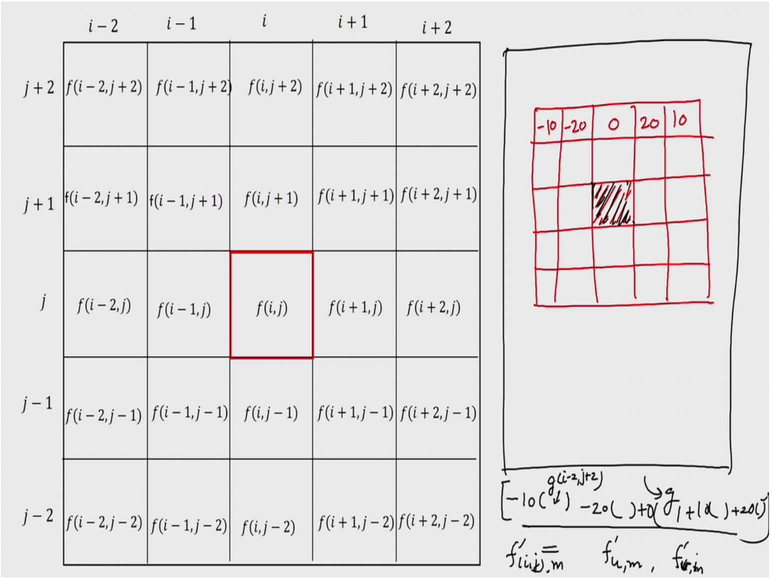
Again now the question is how to calculate the derivative for the Forstner operator ok. In case of image matching we have 2 images and we are running one Sobel operator taking the images from one and to trying to match with other; however, in case of Forstner operator, which detects one point in an image what to do? Let us take only one image and now you are running this Sobel operator over that image. So, what will you do now ok. So, let us draw then image like this, and let us put a Sobel operator at some location this pixel. So, like this ok

I put my Sobel operator here for that pixel now how to calculate the derivative in this case? It is quite different from image matching ok. Let us consider this value of Sobel operator which could be let us say 10 or minus 10 whatever and let us say this is your minus 10 let us say plus 10 and all are 0 fine it could be let us say 8 and plus 8 say I am just writing some values. Now what will do? You will take the value of this image that is this image here and you will multiply with the 10 fine which is just below this section ok. So, 10 into some value plus 8 into some value plus 0 into some value plus minus 10 into some value and you will make the summation for the all the points.

So, all there are 24 points are there from here to here. So, I can say I equal to 24, you are doing all the summation over this point not only that you are doing for 25 pixels including this pixel also. So, this is the way I calculate my derivative fine. So, that is this is a summation at this pixel. Or for this pixel I can say this is the value I call the derivative f dash let us say pixel i j. So, this is the way we calculate the value of derivative in case of Forstner operator. Now I will repeat this process for each and every pixel. So, you can imagine now that I have shifted my operator this Sobel operator by one pixel in x and now I am repeating the whole process and again I am calculating the f dash for i plus 1 j and I will repeat this process let us say for f dash i plus 2 j and so on.

So, this is the way I will do for each and every pixel this kind of operations, and I say that these are my derivatives at different points. So, the moment if there is some distinct point, let us say a point which is quite distinct with respect to the surrounding what will happen? The value of this derivatives will be very very high compared to an homogeneous area you can try it yourself try to take simple black and white image and try to do it you will be surprising that how simple this logics are and, but still they are very strong in terms of their calculation in terms of their delivery of the constructs.

(Refer Slide Time: 42:49)



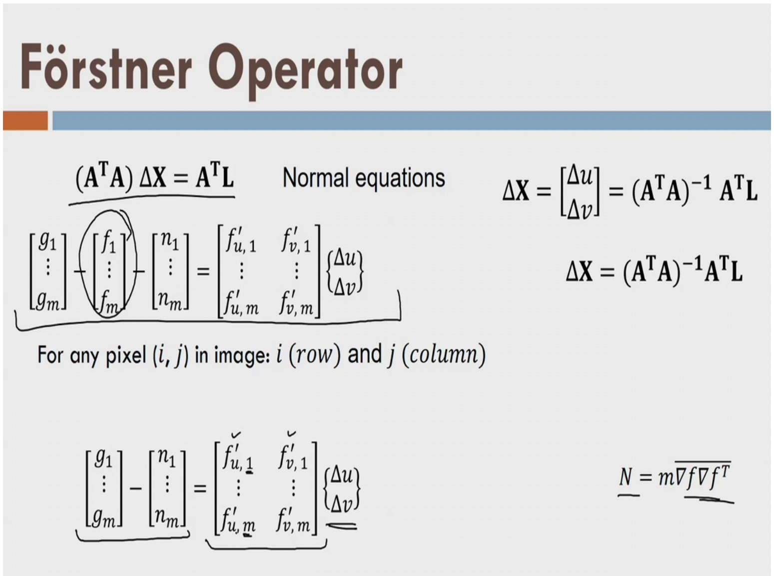
So, let us work again with the Sobel operator and this time again this is f i j and again and again I am keep on filling these values here. So, what will I do this time? Again I will take a Sobel operator and this time how to calculate derivative that is most important now to understand. It is slightly different from the image matching because now we have only one image and this is my Sobel operator right and Sobel operator is same what we have learnt in the last 2 slides before right. So, the Sobel operator is same and now I am saying how to calculate the derivative in this case because its a only one image, its not 2 images that they are matching it is only one image that in which we are trying to detect a particular point which is comparatively different from the surrounding pixels.

So, let us say we have Sobel operator and this is the point where I am right now putting my Sobel operator in the image f or g whatever. Let us say g image I am putting. So, let me draw the g image like this, this is let us say my g image now in this g image I have put it this Sobel operator on this pixel like this now what will happen? Leaving this pixel apart I will first take the values of Sobel operator which is let us say it is giving minus 10 I do not know I am just writing it let us say plus 10 here minus 20 here plus 20 here whatever and say its 0 here right.

So, now, what will I do? I will take the value of g image from here and I will multiply with minus 10 here. Similarly whatever the value of g image here right. So, what will I do now let us see minus 10 into some value of g and let me write it let us say i minus 2 into j plus 2 like this. So, this is value is here and I am keep on multiplying this thing sorry minus 20 minus 20 into again g value and so, on I will keep on 0 into g value plus 10 into g value plus 20 into g value and so, on like this I will calculate; I will make the summation here like this for the whole each and every pixel in this Sobel operator and image g and in this is process is called a convolution.

So, after doing this kind of convolution, I will find out some value here and that will be my derivative of image f or I write it f dash at this pixel location i j right. So, I am writing it f dash at i j for m neighbourhood. So, I can write it m or I can also write it like f dash u at m neighbourhood similarly f dash v at m neighbourhood see.

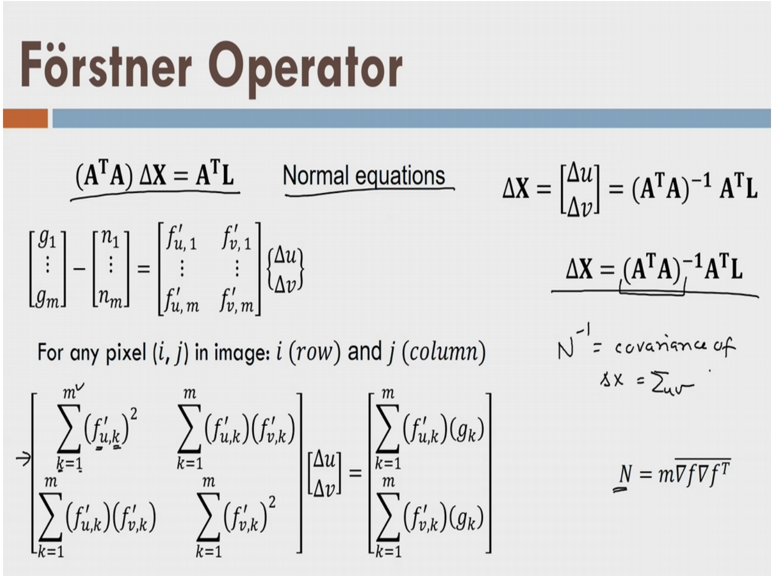
(Refer Slide Time: 45:57)



So, now let us go ahead. Again I will use the same logic here after finding out this one right normally equations and we will do like that. See remember now there is no if you remember this was the function we have written for image matching, but this function is now absent completely here right because there is only one image.

So, I am not writing here this thing it is absent here, and I am writing this thing here like this fine. And I want to find out what is my delta u delta v; that means, I am trying to match my Sobel operator in such a way that the moment it detects that particular point which is distinct from the surroundings, my delta u and delta v should be minimum that is the idea here. So, I have find out my f dash u, f dash v at first location and similarly that m location right. I can you can imagine that we have some kind of n values here in the back slide, and then I can find out like this ok.

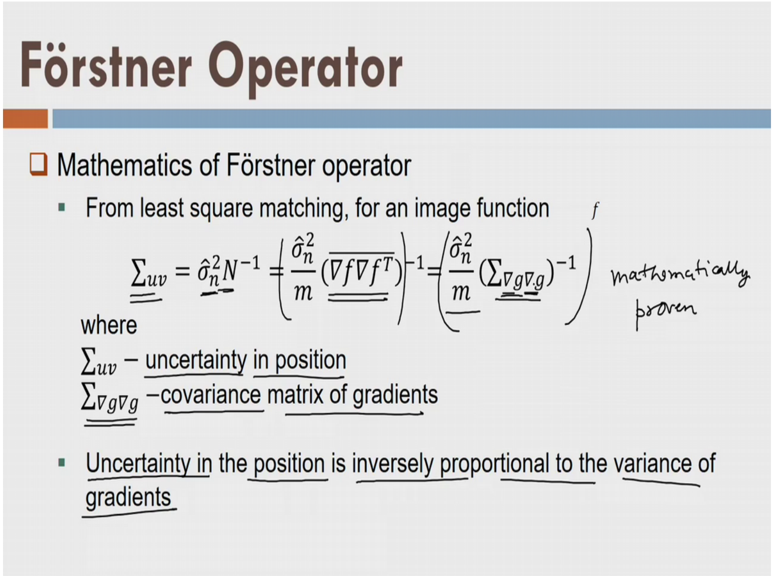
(Refer Slide Time: 47:00)



Then in the Forstner operator this is the logic again I am using it normal equation and now I can write this thing.

So, there if I do this A T A this my A T A will come like that; that means, I am multiplying my all f dash for our k and where k is equal to 1 2 my neighbourhood and remember this is quite different calculation and then the image matching. So, it is depends on how do you calculate f dash fine. So, this is my delta x here again the solution by least square solution and there I have this thing now what about the end matrix which is nothing, but A T A and we know that N inverse is my covariance of delta x; that means, sigma u v.

(Refer Slide Time: 47:52)

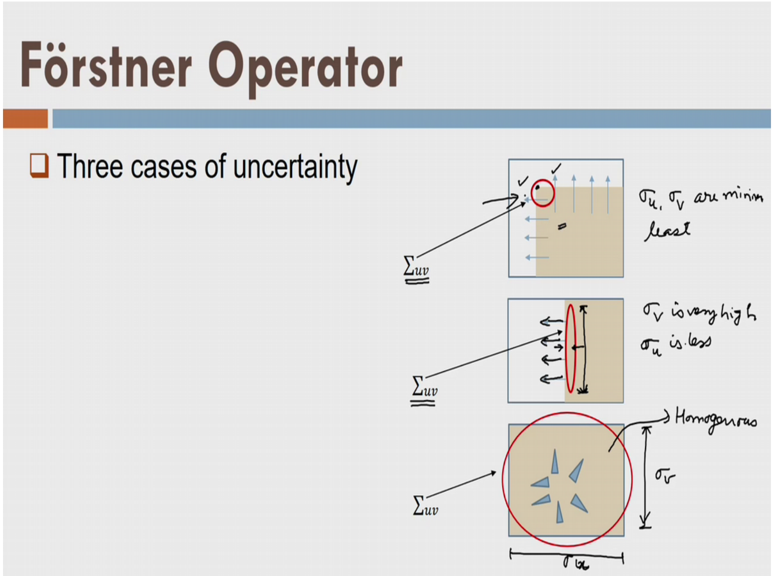


So, I can write my sigma u v is equal to N inverse time some reference variance and I am writing this matrix A T A inverse like this delta f by delta f T.

And like this. So, it is nothing, but it is proven well that this equal to this which is my delta g g is nothing, but the covariance matrix of the gradients remember gradients in my original image and these are the uncertainty in the position; position of some pixel that is the that I want to detect. So, this is my uncertainty. So, now, we can see here, but it is mathematical proven this logic. So, I can see here the uncertainty in the position; that means, uncertainty in the detection of the position of point, which is distinct with respect to surrounding is inversely proportional to the variance of gradients; that means, higher the gradients is it is very easy to detect the point.

And that is why the moment we say it is a distinct point and; that means, it is quite different from the surroundings, it is very easy to detect that point in an image by Fostner operator. Why because, the values of my gradients of that image around that point will be very high because of the difference between the popular point pixel with this neighbourhood and that is why it is my logic its working there ok.

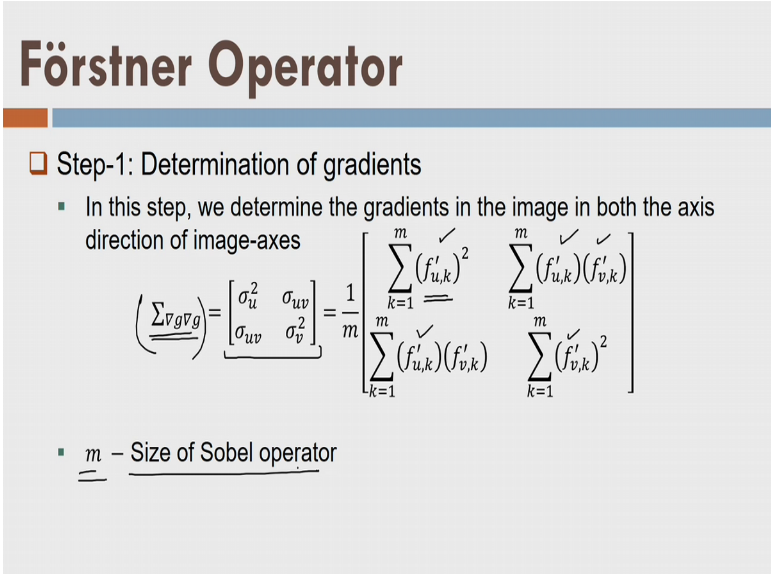
(Refer Slide Time: 49:29)



So, let us see what is the graphical meaning of this there are 3 cases of uncertainty here you see that it is very less here in this case. In this case you can see the uncertainty in y is very high and in this case both uncertainty in my x that is I can say sigma u and sigma v are very very high right.

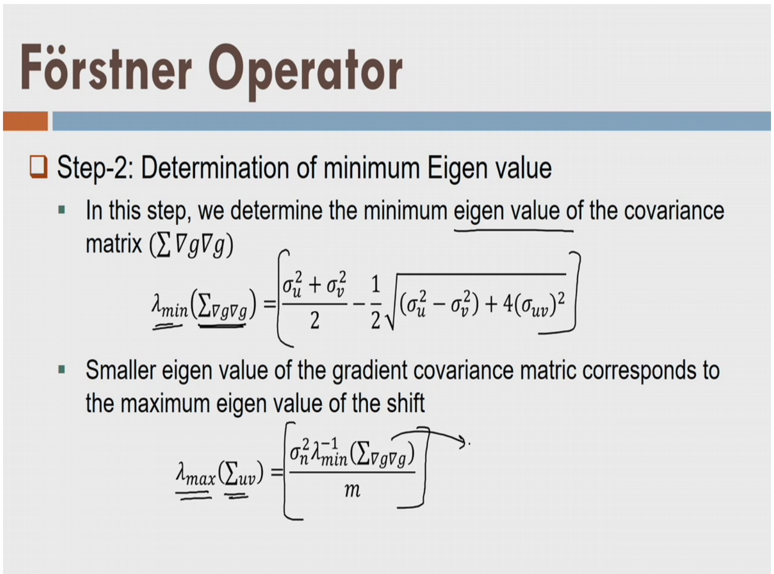
Because of the homogeneity of the area this is my homogeneous area; that means, there are no gradients here on the edge we have. So, this one this x direction is very small where as my sigma u less and here you can see sigma v is very high, but the moment this situation here both sigma u and sigma v are minimum or less least I can say. So, this is the point easy to detect by the Forstner operator right and that is why this is kind of situations we want; so, that we can easily detect these points very easily right, because it is quite different from its surroundings. So, here it is white and here it is some other colour fine.

(Refer Slide Time: 50:52)



So, now, how to detect what is the process. So, now, we are looking for what are the steps we should now we have understood a logic, now we are looking into the steps. So, this is my delta g delta g which is I am saying equal to this fine that we have already learnt. So, now, it is equal to this matrix. So, I will calculate only these values f dash and I will just put in this matrix, I construct this matrix then here my m is size of this Sobel operator remember how to calculate the f value ok.

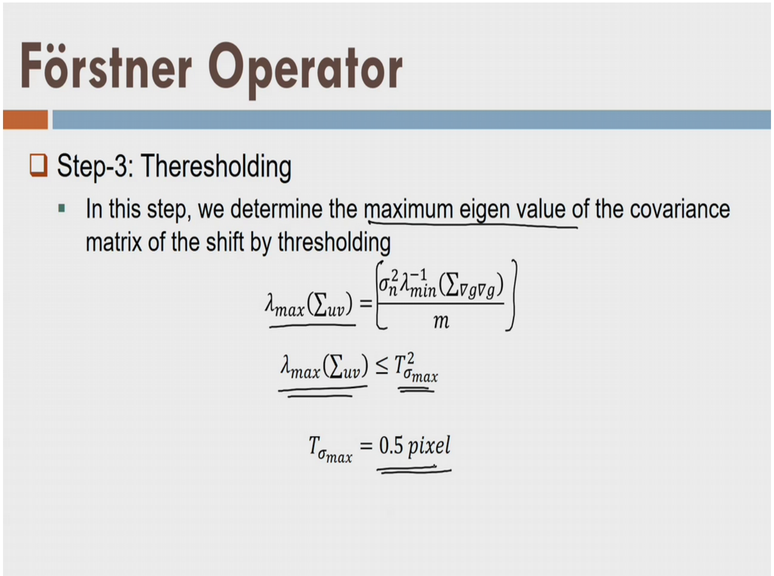
(Refer Slide Time: 51:27)



There I will calculate the eigen values of my delta g delta g, which is given by this logic remember again eigen values and eigenvectors. So, these are my eigen values. So, that is my minimum eigen values similarly what is my maximum eigen values. So, like this ok.

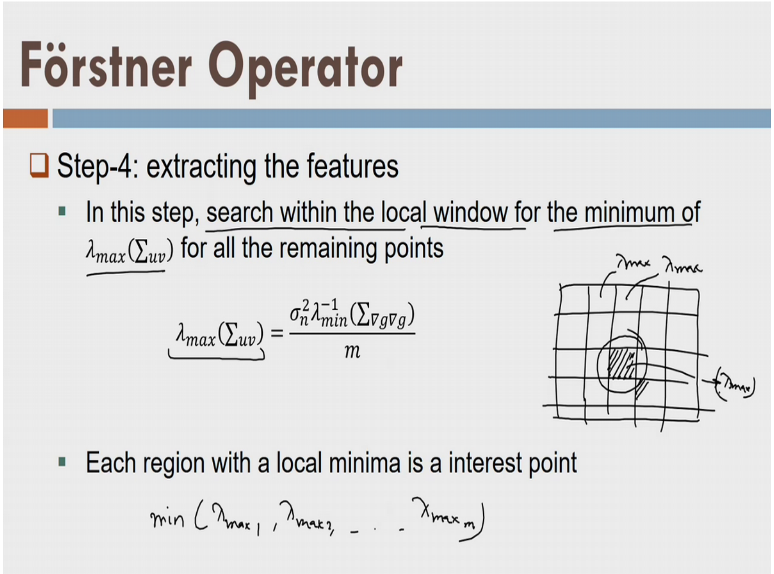
So, the smaller eigen value of gradient covariance matrix correspond to the maximum eigen value of the shift fine; that means, because they are inversely proportional. So, I can say that the maximum value of this one is the minimum value of this one and this is my minimum value here right.

(Refer Slide Time: 52:08)



Ok now find out the maximum eigen value of the covariance matrix of shift by thresholding; that means, I have already find out this. Now put some threshold value let us say something like this then they will you put 0.5 pixels, and then by this thresholding I can find out where the point is which is distinct in nature that is it is distinct with the surroundings I can find out that point is very easy you can understand what are the steps.

(Refer Slide Time: 52:39)

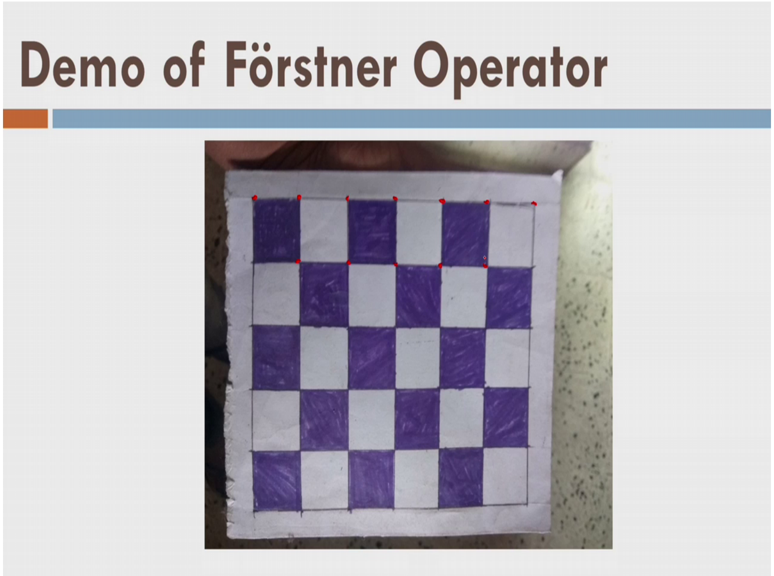


So, there now step 4 is there extracting the features how to extract the features finally, and this is the last step.

So, we search within the local window for the minimum of this one; that means, if I have some window like this and different points I have calculated all the values of lambda max, let us say this is my lambda max this is another lambda max and so, on because let us say this is the point which is quite distinct or this is the point quite distinct fine. So, around that all lambda max will start coming now what will happen? I will take the minimum of the lambda max; that means, I am taking the minimum of let us say lambda max 1 then lambda max 2 and lambda max m because m is my neighbourhood here.

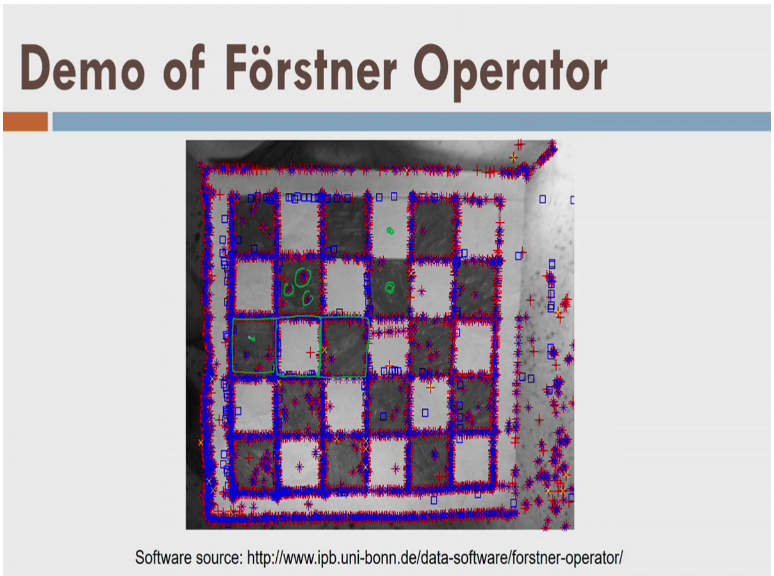
And the moment the lambda max is minimum for a particular point, I will declare that this is the my point that is final point final answer; that means, probably I am going to go here because lambda max will be minimum here. So, this is my point that is the I want to detect in my image. So, this is the logic of Forstner operator I hope that we have learnt today enough material about image matching and let us see this is a demonstration of my Forstner operator.

(Refer Slide Time: 54:08)



And again we are putting this image where I want to detect these points these point remember because they are distinct from their surroundings, and now if I run the Forstner operator for this image what will I get I will get these points as my distinct points.

(Refer Slide Time: 54:25)



Now you can see here there also I am getting a lot of points I am clearly tell that these are my edges. So, these are my edges here clearly tell by Forstner operator fine now I can detect my features like this. You can see in the within this homogeneous area there is nothing is detected nothing is detected, but there are some noise is there, because this was made manually by some student right by some person since this image was created by you know filling up the boxes by colour. So, that there was some a non-uniform filling here. So, it detected some points like this within that area.

And then that is the way we detect on the Forstner operator detect the distinct points. I hope this lecture was very useful it could be little complicated for you at this moment, but try to learn it. It is very useful for g c p edges and digital image processing. So, here we can say that now I can detect the GCPS also in the image first my automatic point detections using Forstner operator ok. So, we stop here and then we will meet in the next lecture on the closed range photogrammetric and that will be the last lecture in the module.

Thank you.