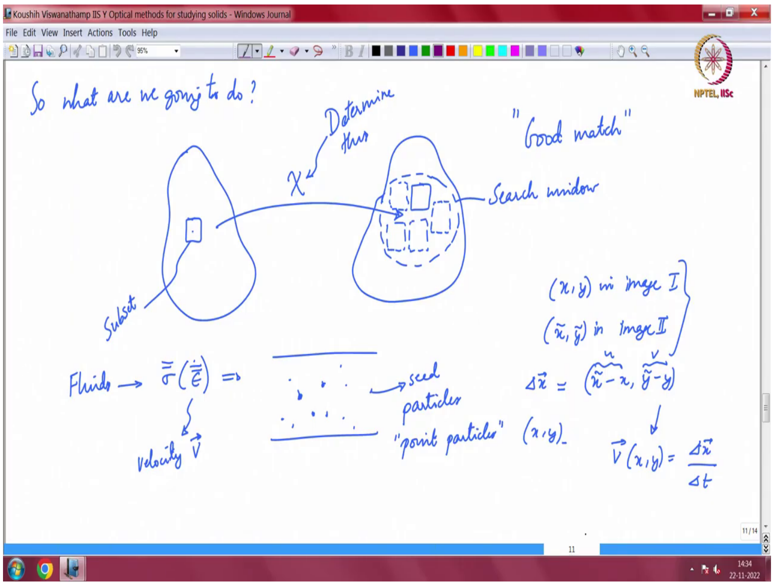
**Optical Methods for Solid and Fluid Mechanics**
**Prof. Aloke Kumar and Koushik Viswanathan**
**Department of Mechanical Engineering**
**Indian Institute of Science – Bangalore**

**Lecture - 26**
**Iterative implementation of DIC**

All right welcome back. So, we were discussing formulation of DIC and we were going to derive the equations for iterative scheme for determining the displacement fields and its gradients for a solid deformation. Before we start just to make sure that we are all on the same page and we are up to speed dial just backtrack a little bit and continue from where we left off.

**(Refer Slide Time: 00:24)**



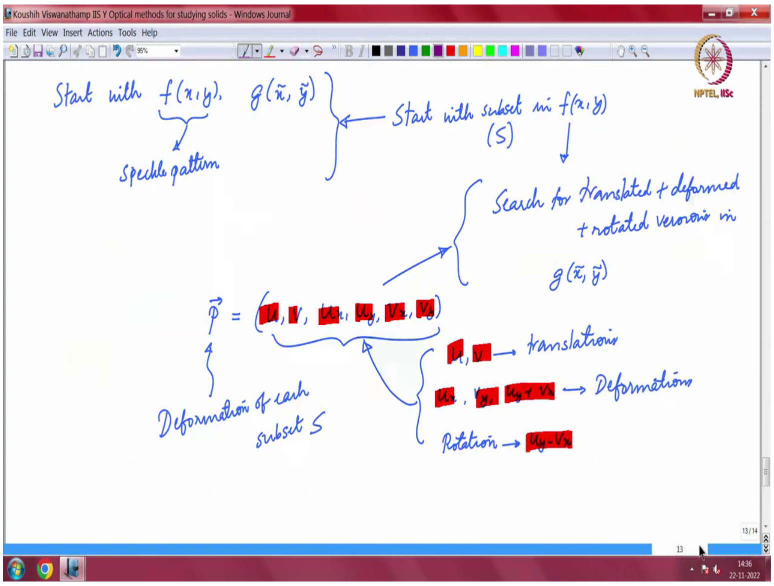

So, this is where we were we had discussed this idea of undeformed configuration on the left and a deformed configuration on the right and which said that the aim is to determine this mapping called which we called Chi which takes the undeformed configuration to the deformed configuration. And we to do this we take a subset at a particular point in the undefined configuration that is marked over here and we search within a search window in the deformed configuration and we try to find a good match.

Again I put this good match within codes because if you recollect we discussed that for fluids the only thing you need to match is a rigid translation for the subset because points are all moving around. But in the case of solids you have this additional problem that the section can be deformed the subset can be deformed by itself which means not only is the rigid body displacement important which we have denoted by u and v u being the horizontal displacement v being the vertical.

But also you have to worry about shape change. So, which means the gradients of u x u y v v x v y and rotations right. So, this Square can rotate by some arbitrary angle at each particular point. So, you need to. Now look at six parameters when making the search from the undeformed to the deformed configuration. Now this makes the cons the search a little bit more complicated because of six variables.

But additionally it also puts constraints on the image right. So, you cannot have any type of image in which you can get a successful search and we discussed two of these problems I had introduced this reference by stair motion.

**(Refer Slide Time: 02:16)**

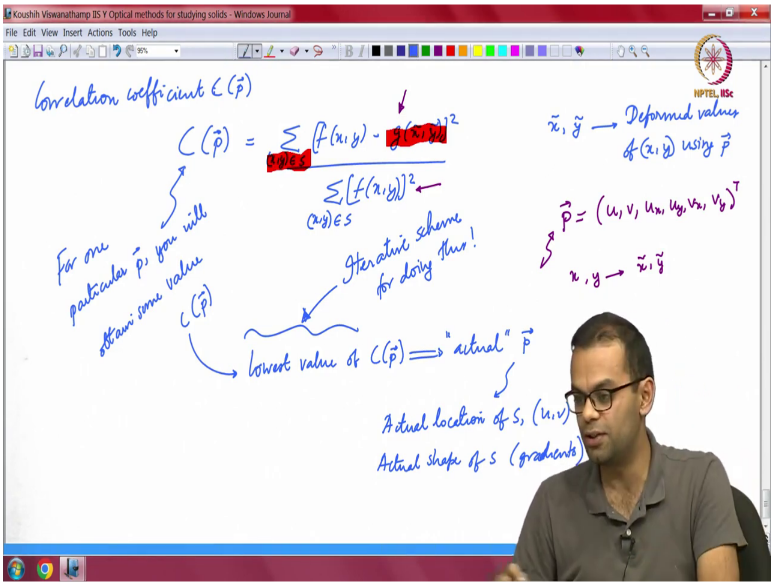



And I showed you a couple of problems the corresponding correspondence problem and the aperture problem we discussed that also in the last session and just to make sure that our notation is clear this is what we had used. So, f was a undeformed image g is a deformed image and we also realize that you have to use a speckle pattern in the image and then follow the speckle as it deforms you cannot have an infinite lattice because then that will lead to a correspondence problem you cannot have infinite lines because at least one aperture problem.

Now to do this search you start with the subset inside f like I mentioned here and you look for translation deformation and rotation. Which means translations like we saw or u and v deformation which is basically the strain has three components in two dimensions. So, that is u x v y which are the tensile or the stretch or compression components and u y plus v x which is the shear and there is also a rotation component in 2D there is only one rotation component.

So, you have six unknown variables u v u x u y v x v y and you have six physical operations that are potentially performed to a particular subset s. So, that is that uv are the translations u x v by u y plus v x are the shear plus stretching slash compression we generally call them deformations and you have the rotation u y minus v x. So, six represented in terms of six all right.

**(Refer Slide Time: 03:45)**




So, the heart of this entire scheme of course is this guy called the correlation coefficient which we have denoted by C sometimes called Corr. So, notation might vary and it is defined as the follow as the following. So, you take the search region which is s the subset sorry subset which is s and for every Point location inside s you evaluate f at x y that particular Point minus g of x tilde y tilde now x Delta y tilde are deformed values of x y.

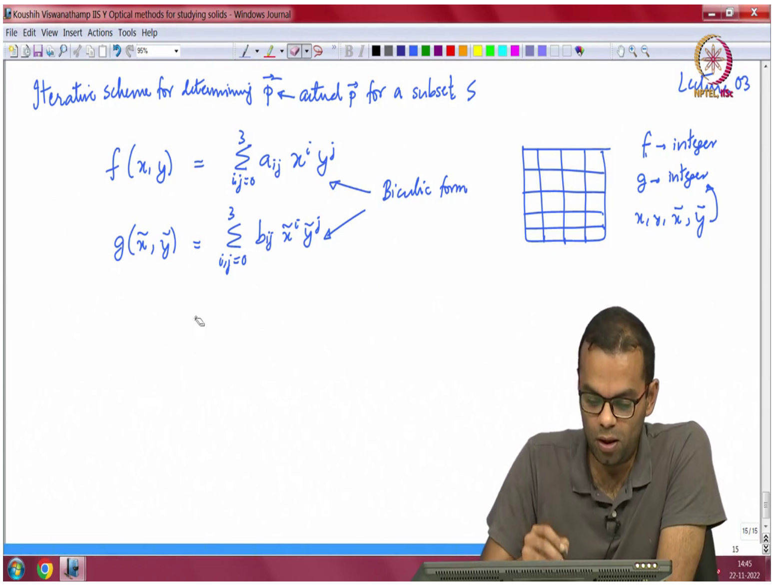
So, you have let us say some value of P being again this. So, this is your six Dimension I am going to use the column Vector notation for this just to be consistent it does not matter you can flip transpose everything I am going to show and still be consistent if you wish um. So, this is the six component Vector p and x tilde y tilde is obtained by adding p to x, y. So, you have x, y you do Taylor series expansion you will get x tilde y tilde.

And the first two terms in the Taylor series expansion are the translation terms and the gradient terms. This will work out the algebra today but at least I am just giving you an idea before we start. Now the whole point is you do this let us say for some arbitrary p you could think of this being a six dimensional space for instance with axis u v u x u y etcetera. And pick various values or six tuples for this Vector P add it to x, y and get x into y tilde.

And then find out this sum this difference and square it and divide by F this is again just done for normalization it doesn't really matter usually and then you find a c. So, for one particular value of P bar you get a c you can do this for every single value of P bar you can change u by a little bit you can change u by a little bit change u x by little bit blah blah and get different values of P bar cut different corresponding values of C.

You collect this entire set of values of c and then you look for the smallest value of C or an extremum value of C and at that value of C the corresponding P for which you get the smallest gives you the actual P that is the idea. So, the correlation is highest when you do this type of a thing. So, we will continue from here I will do the formal derivation little bit to evaluate this iterative scheme that we mentioned. and we will see how you can do an implementation based on this iterative scheme.

**(Refer Slide Time: 06:30)**




For determining the final P, so, this is the actual P bar for a subset so, subset s. Now remember this is a function of the subset. So, for different parts of the image you can get different values of P which means that you will naturally expect that if the deformation is not homogeneous various values of displacements and its gradients at various locations and this entire iterative scheme is for one particular location.

So, you typically will start with a grid and iterate do the scheme at every single point on the grid using a window subset and a search window for every single point. I will show you an implementation again in the next session hopefully but today we will set up some of the structure for getting there. Now before we develop this scheme we will just recollect one small bit of information that I also introduced last week which is the fact that f and G as I will show you are represented in the form of a bicubic form okay then interpolated.

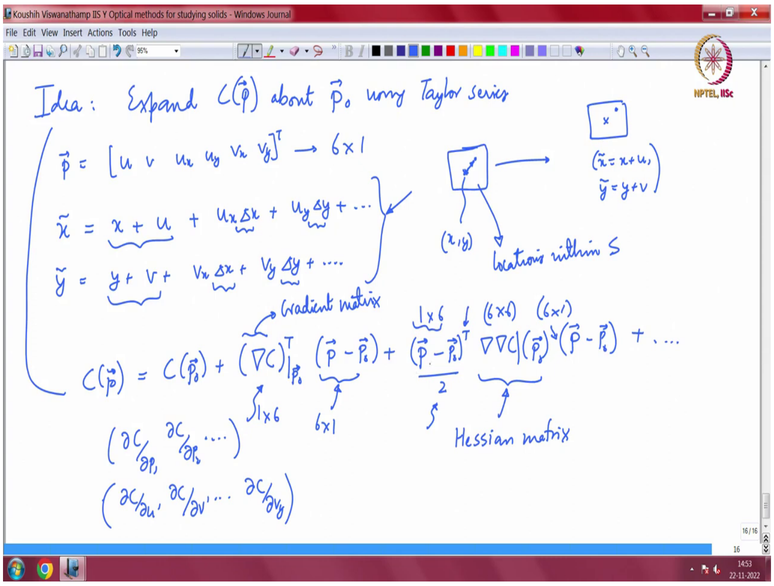
So, we discussed this at length last time likewise for G x tilde y tilde R coordinates in the deformed image and this bicubic form allows us to take derivatives. So, for example if you remember in the original image you typically have x y as pixel location. So, they're all integers and the intensity if you have an 8-bit image the intensity goes from 0 to 255 or you know 10 bit it goes from 0 to 1 0 to 3 and so on.

And so, typically f is an integer valued function g is also an integer valued function as our x y x tilde and y tilde okay these are also integer valued because pixel locations are not integers. So, by doing this bicubic interpolation it allows us to take derivatives second derivatives specifically as we will see and makes them also continuous. So, second derivatives can be defined very easily and you will not lose data as you go taking higher derivatives okay this is something to keep in mind.

So, all the gradient Expressions that I will write down they're all with the view that you have this bicubic interpolation underneath okay. So, for instance you can actually evaluate if you want to evaluate dou C by dou p i right p i being one component of the vector p you can easily do e c by dou x 2 x by dou P I using the chain rule for instance or dou C by dou x tilde dou x tilde by dou p i you can do plus dou c plus dou y tilde dou y tilde by dou p i and so on.

So, you can do these types of chain rule applications without worrying too much about whether your image has enough resolution and so on. if you have this by cubic interpolation anyway we will see this as we go along okay. So, the long and short is the following the idea that we are going to use.

**(Refer Slide Time: 10:05)**




The main idea is to expand C of P about some point P naught. It does not matter what point this is just an arbitrary point will make some sense out of what point to use at the end.

And we will do this expansion using Taylor series simple Taylor series truncated at two terms okay. Now remember p is a column Vector as I as I mentioned before. So, please keep this notation in mind.

So, it is a six by one Matrix six columns by one row I am just using this notation. So, that we are consistent in all the Expressions we write down. So, this is our P Vector the p is useful for writing x tilde in terms of x and y tilde in terms of y. So, for example this is this is typically what extent I will look like okay. So, if you think back again from a discussion about the difference with fluids the first term here represents a translation.

So, if you add a square this is the center of the square and this square is in the undeformed image at the location x, y the first term alone right. If you just keep the u and v terms it means that the square is getting moved translated rigidly translated to a particular location x tilde equals x plus u y tilde is equal to y plus b. Now in addition to this the second term has this Delta x Delta y that is dropped in.

So, it is no longer just dependent on the central location but it also depends on locations within the subset okay. So, if you have some point that is Delta x, Delta y away from the center that point will correspondingly move to another location here given by these terms okay. So, you have a stretch or you have a rotation that is what this means right. So, this

Taylor series this one is just giving you the stretch this one's giving you one component of the shear plus the rotation and this is analogous.

Of course you have higher order terms that will include second derivatives of the displacement and so on. And you will have more components to worry about but we will just assume that since we have a six component Vector P we will assume that these are the only terms you have to worry about okay all right. So, in order to determine this I said we will expand C of P about P naught.

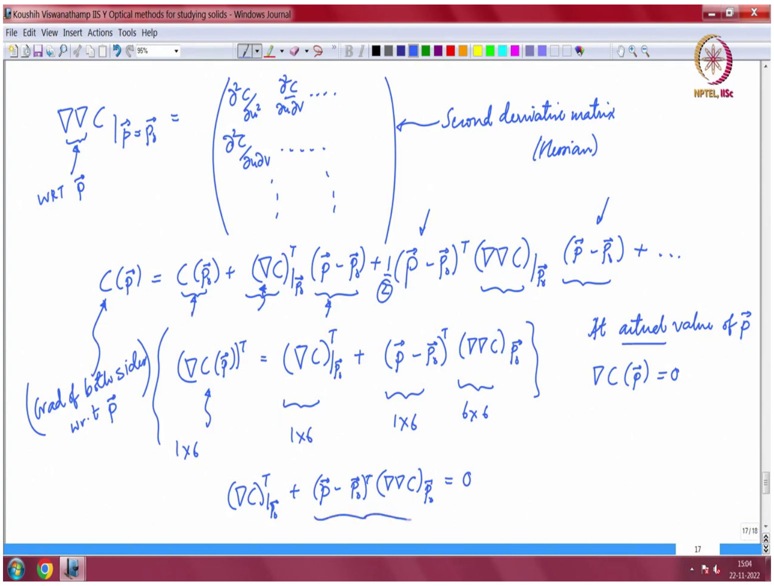
So, the expansion goes something like this the first time across the C of P naught and the second term is this. So, this is the second term in the Taylor series this is similar to if you had a function f f of x and explaining about x equals x naught you have the first term f of x naught in second term d f by d x times x minus x. Now this is like that term okay. Now notice that grad C rather P minus P naught is a six by one like we have here above.

So, it is a column Vector this guy is a row Vector. So, it basically has dou C by dou P one P one being u dou c by dou P two and so on.. So, this is nothing but dou C by dou U dou C by dou V and so on. dou C by dou v y that is what this is and then you have another term we will keep two terms you will see why we will have another term which will be the second order term in the Taylor series.

So, that will look like this and higher order terms okay which have larger additional derivatives on C this is called the Hessian matrix this is called the gradient Matrix okay and just this factor of two of course is the two factorial that you get in a Taylor series typical Taylor series expansion. So, the next one will have a factor of one by six and so on three factorial just to be sure this being a one by six.

This being a six by one will give you a scalar you have a row Vector times a column Vector here likewise this being a one by six this will have to be a six by six this will be a six by one again you will get a one by one which is the scalar right. So, you should have Basics six by six six by one for this. So, notice there is a transpose here. So, this P bar minus p 0 bar transpose will be a transpose of this which means the Hessian is a six by six Matrix.

**(Refer Slide Time: 16:09)**




And it is not very difficult to guess its form is basically this second derivative of C right. So, this one will be dou square C by dou U dou dou V blah blah blah blah and so on. So, these are all second derivative Matrix. So, you have C of P this note is also this is evaluated at P equal to P naught. So, please remember that and this is again I am just writing down what we had on the previous slide this is again at P equal to P naught plus blah blah blah.

Now this gradient is gradient with respect to P just like this second derivative with respect to P right it takes with respect to u and v and so on. So, this is with respect to P this is also probably evident in the expression we wrote down for the grad p a grad C sorry. So, the ingredients again are evaluated at a particular part. So, they are constants. So, I will just take a gradient on both sides of this with respect to p and transpose the entire final result.

If you do that the first term will just be grad C of some arbitrary p transposed the first term on the right hand side will be 0 because it is a constant remember this is a constant evaluated at P equal to P naught. So, that will go away if you do the same thing on the right hand side the second term this again a constant it will come out and then this has only P bar minus P 0 bar. But since you are multiplying through a matrix and you are taking a transpose gradient of that with respect to P will give you just identity right.

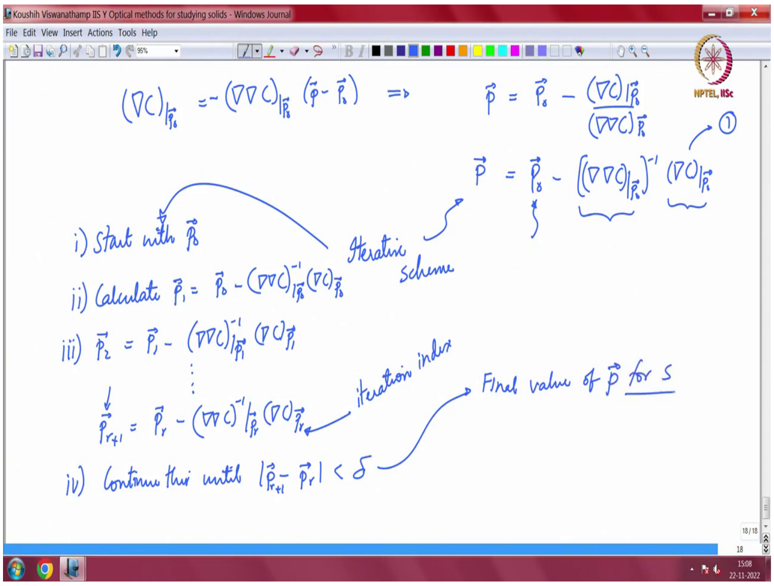
So, then you will just get this which is again grad C at p 0 bar transpose and if you continue this process to the second term again you have a constant sitting in the middle and you have a term on the right hand side. So, I take a transpose of this entire thing take a gradient like I said before the gradient p of this term will give you identity again gradient P of this term will give you another identity of a factor of two.

So, the two of them will add up to give you just one term. So, you will get p 0 bar minus p 0 bar ok. So, this is if we go back you will see that we had defined gradient C transpose as a one by six Matrix. So, this is one by six this is of course one by six P bar was a six by one. So, you should have a transpose here. So, this will be a one by six and this will be a six by six. So, the whole thing will give you one by six.

So, it is just bookkeeping there is nothing deep or sophisticated about this it just make sure that you have the same consistent notation throughout there is nothing else to remember okay. So, if you do that. Now we have this relationship and if the at the actual value of P remember we said that this gradient of C is 0 because C is an extremum. So, if you actually have the displacement slash Shear slash stretch slash rotation of that particular subset.

Then the value on the left hand side should be zero which means that this grid C on the right hand side should give you zero okay it should add up to zero. Now you can shift the second term on the right hand side here.

**(Refer Slide Time: 21:27)**




And rewrite this expression whereas and take a trans you can take a transpose again and it does not matter on both sides if you go back I had a Hessian term on the right hand side. So, I will have I will have an expression like this the negative sign okay. So, from here I can write down P bar this P 0 bar Plus sorry rather minus grad C p 0 bar divided by I am going to be a little bit loose with my notation divided by the Hessian.

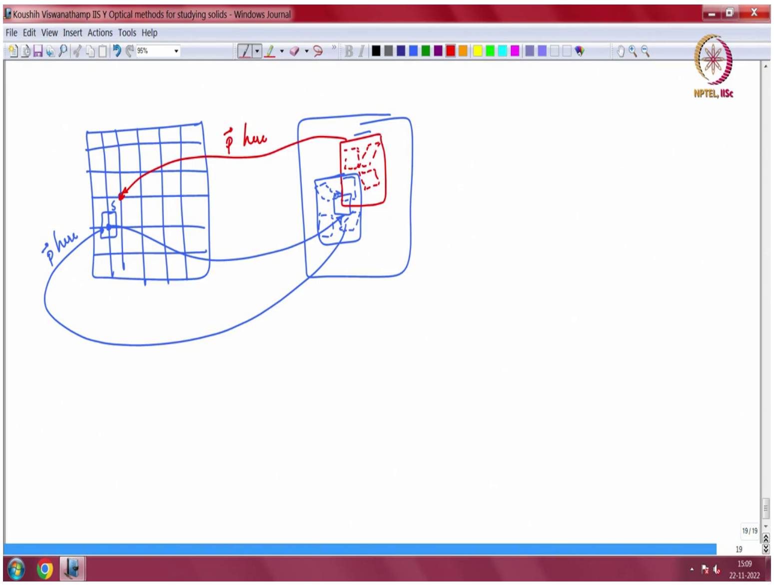
You actually have to do an inverse because these are matrices. So, maybe we will write it out in the correct form. So, this will be. So, this is basically the only equation you will need for the iterative scheme. So, let us say you start with a value of P, P naught this is your initial guess. So, to speak for the entire six component vector you can just evaluate these guys because all of them are done at C you can evaluate these guys.

And then get a slightly better value for p okay it still will not give you the minimum because if you evaluate C of P after one iteration typically you will not get the actual minimum. So, you will get another value which is slightly better than the first guess. So, then you take this and you iterate. So, this is. So, you start with P naught let us call this equation one and then you calculate P1 is equal to P 0 bar minus the Hessian at P naught inverse times the gradient at P naught and you take this and you repeat it.

So, you calculate P 2 P 1 minus. Now great at C inverse at P 1 graph three at P 1 and so on. So, in general after the Earth step you will have P R plus 1 is P R minus this Hessian evaluated at PR again multiplied by the gradient evaluated at PR and notice also that this subscript I am using here is actually an iteration Index this is not an individual component I am using a vector notation this is not an individual component of P it is the iterating index and you do this as a purely Matrix equation.

So, you will have a column Vector on the left and you have a column Vector on the right which is the sort of Correction term after every iteration okay and you continue this until you get some conversions right until you get you until your additions become. So, small that the value by which P changes after the Earth iteration becomes very negligible. So, at that point you can stop the iteration and the final value this gives you the final value of P. And remember this is for the particular subset s OK where this is all performed.

**(Refer Slide Time: 25:29)**




Now this is a fairly simple scheme. So, typically what you will do is you start an image you have a grid or an image I will show you some of this implementation in the next session and at each grid location you have a subset you look at the deformed image in a search region get this upset out and map it to various locations in various shapes and so on.

That is essentially what you are doing by doing this iterative scheme and depending on where it matches the best where you get the lowest value of cop from the iteration scheme that is the value of P at this node. And then you move on to the next node you have a for example if I made this node red I have another search window which could potentially overlap with the previous search window where I am doing the same type of search and I get another value for P bar here.

So, that is the idea very simple in principle and once you are done you have a grid of values for the vector P. So, that gives you the full field displacement and deformation a displacement gradients. So, that is basically a you know the iteration scheme that underlies most DIC implementations. Now of course there are nuances in the actual implementation there are things you have to take care of for instance you have to do a little bit of pre-processing things like filtering masking and so on.

And you have to do a little bit of post processing for interpolation and things like that. So, what we will do in the next session is I will walk you through a sequence of steps with a couple of sample images I will show you what each step does how the iterations scheme progresses and what you will get at the end. So, that you can perhaps use this as a starting point for your own implementation and once that is done.

We will also have a demonstration session showing you how to use an open source code for doing many of these calculations if you are not up to writing your own code.