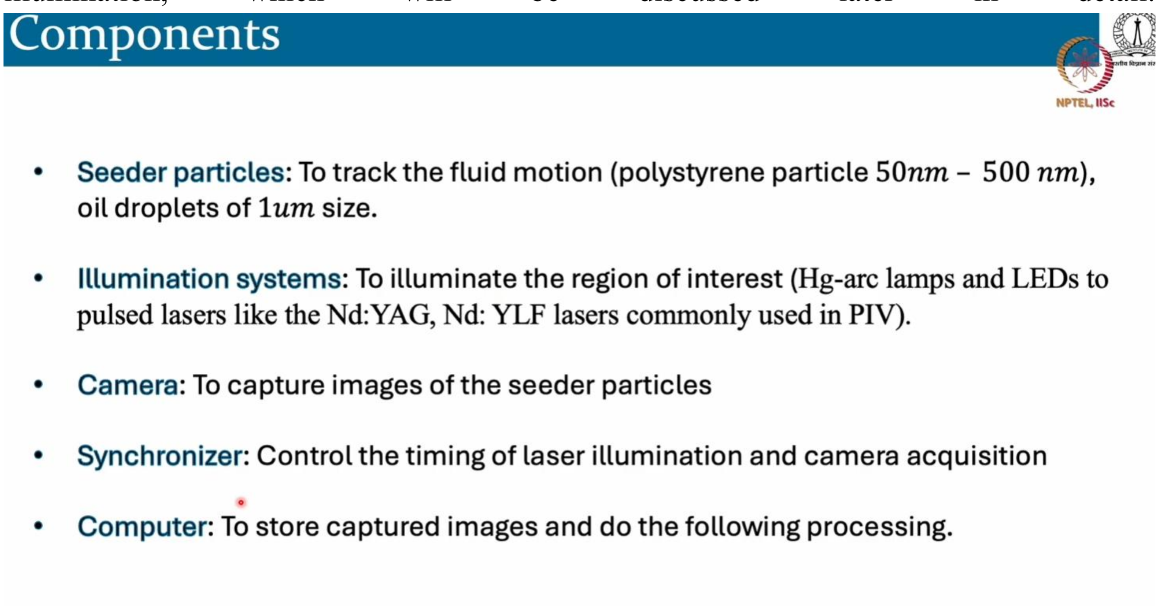
Advanced Measurement Techniques in Fluid Mechanics and Heat Transfer
Prof. Saptarshi Basu
Department of Mechanical Engineering
Indian Institute of Science, Bengaluru
Week – 08
Lecture - 39
Micro PIV – 2

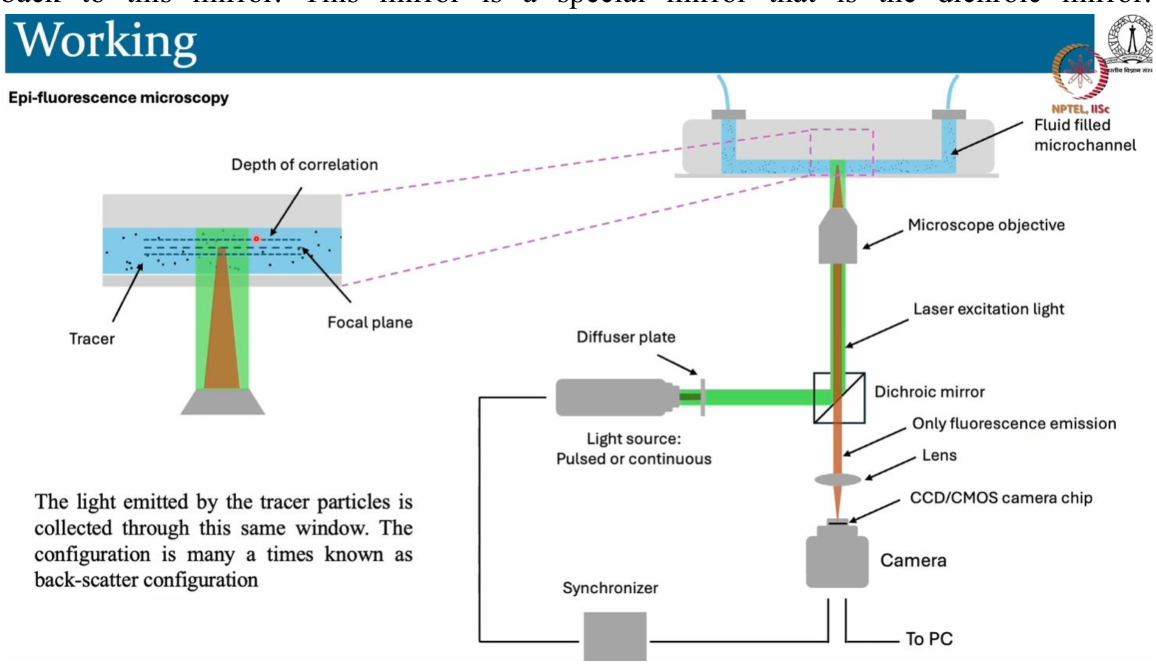
We will start by discussing a bit about the components of a typical micro-PIV system. Although most of it has been discussed by Professor Basu in his lectures on PIV, for completeness, we will just discuss it in brief. So, the first and most important thing is that the cedar particles are used to basically track the fluid motion. Fluid can be anything, gases or liquids, as we have discussed, and we will be discussing a lot more about it in the upcoming lectures, so this is one of the most important components, and it has to be chosen very wisely. For example, you can choose polystyrene particles of very small nanometers; it can go up to 1000 nanometers also, and you can also use oil droplets of 1 micron size. Then the illumination system is basically used to illuminate the region of interest, and in the case of micro PIV, we are more concerned with volumetric illumination, which will be discussed later in detail.



So, this is very important, and there are a lot of illumination systems; however, lasers are the most widely used illumination systems because of their ability to emit monochromatic light, and they are highly collimated with very high energy density. One of the advantages is that we can easily turn a laser into sheets for recording, but in our case with, say, micro PIV, this sheet doesn't work, which we will discuss later. The usage of semiconductor lasers like ND YAG and ND YLF is very common, and they can actually deliver a large source of light energy in just a few nanoseconds, which can help capture images of very fast flows even in microchannels. To capture images of the cedar particle, we obviously need to have an appropriate camera system, which can be a high-speed camera.

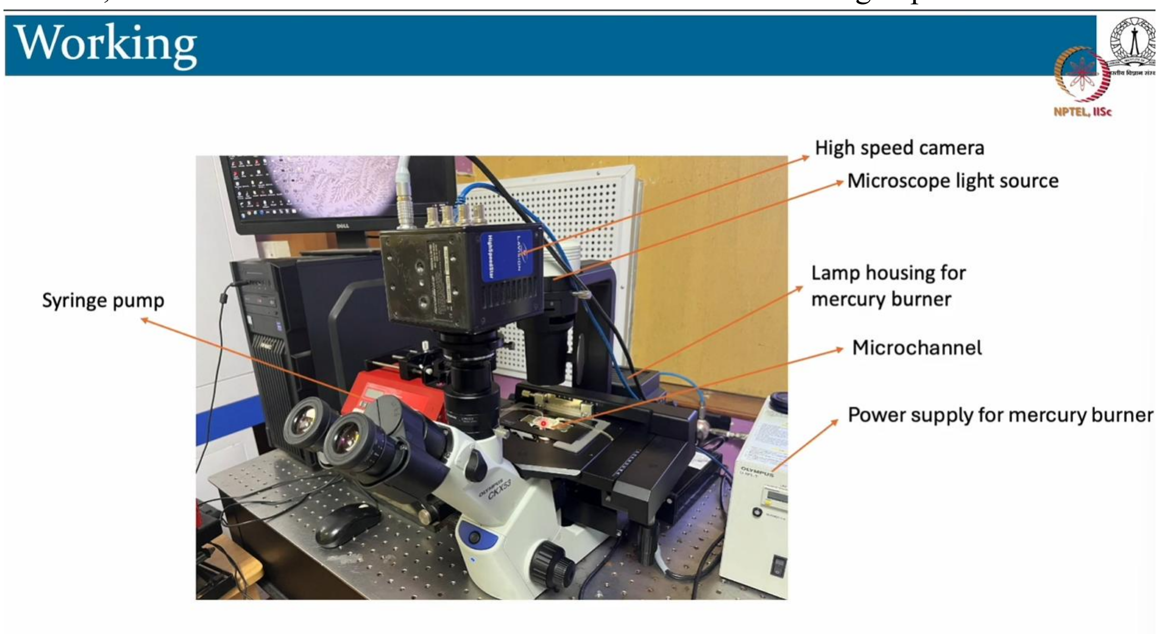
The synchronizer controls the timing of laser illumination and camera acquisition. And in our case, we mostly use a PTU, that is a programmable timing unit. And finally, you should have a computer or something where you can actually take the data, store the data, and ultimately take it to the processing part. Now, we will discuss briefly the working of a typical micro PIV. So, this is how a typical micro PIV setup looks like.

So, if we start from this light source, you can see that this light source can be pulsed or continuous as needed, and the capability of the light source is obvious; you can see a laser passing through a diffuser plate, and this diffused laser light falls on this mirror, and this light reflects at this angle through this microscope objective, which you can modify as needed. You can see this microchannel; this is the inlet, this is the outlet, and you can also see these particles, which are basically fluorescent particles. Now, when this green light falls on the fluorescent particles, it emits back the fluorescent signature, which is, say, the red light. So this red light is again emitted through this system, and now it comes back to this mirror. This mirror is a special mirror that is the dichroic mirror.



A dichroic mirror has different properties for different kinds of wavelengths, so mostly they are used for filtering and splitting or combining lights. Here, it is actually filtering, and it is not reflecting this emitted light; instead, it is actually letting it pass through. see you can see only the fluorescence emission is coming out through it and it goes through a lens and then it directly hits the camera chip the sensor and this is your camera and this camera and this light source is is basically controlled by a synchronizer as discussed previously and the data from the camera goes to the pc so this is basically a typical working setup of a micro piv This kind of micro PIV is also known as epifluorescence microscopy, and so the main principle is that the light emitted by these tracer particles is collected through the same window here, and the configuration is many times known as back scattering configuration. So, now another interesting thing is that if we zoom in on this region, you can see that we find a focal plane, but we are actually illuminating a volume. So, there is something known as the depth of correlation.

So, we will be discussing a bit more about it in future slides, but to give a brief idea, within this depth, basically, the particles that are illuminated contribute significantly to the correlation peak, and beyond that, we neglect the particle illumination and ignore the signatures coming from the particles. So, this is another setup where you can see it is basically an alternative to the micro PIB that I showed previously, where you can also use an inverted microscope to do the same kind of PIB. Now, here you can see one of the inverted microscopy setups with a microchannel sitting in this place, and this is the power supply for the mercury burner, and this is the lamp housing for the mercury burner. We are using this mercury burner to generate fluorescence light. Then you can see the microscope light source here from where this yellow light is coming on this micro channel, and then we have a high-speed camera.

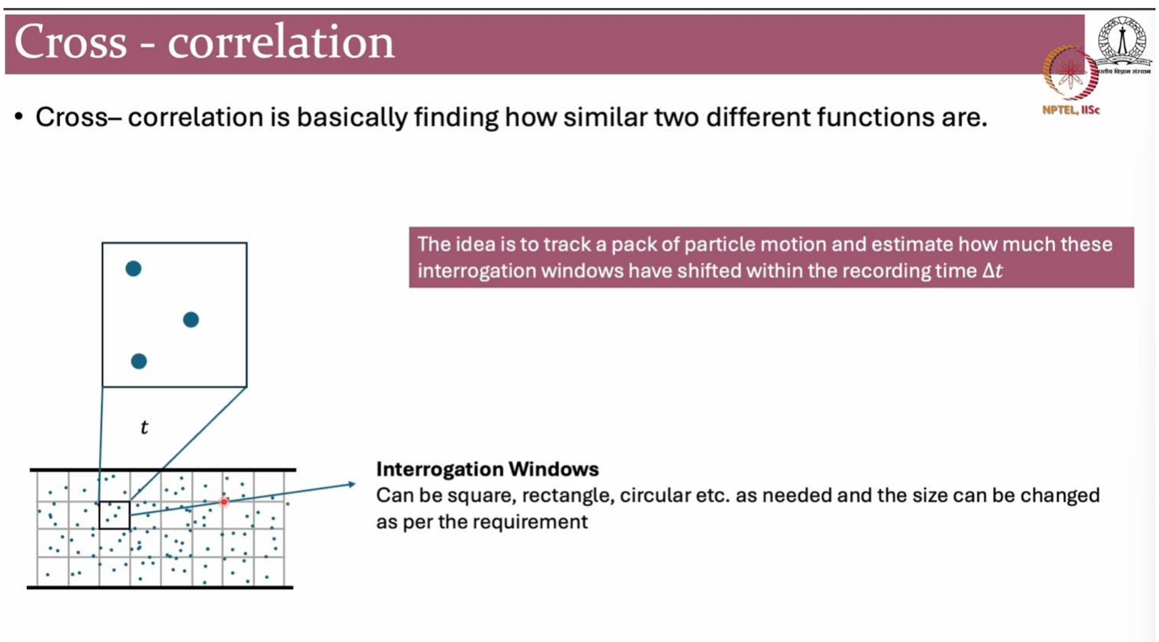


This is a UX Mini 100 high-speed camera which uses a CMOS sensor, and it is very compact and light, and this is from La Vision. Then you can see a syringe pump on this side, so when you actually... To make the flow inside microchannels, you can seed the particles using a seeder particle, and you can actually record at whatever rate you want by using this camera, which is connected to this PC.

Then you can take out the data and process it somewhere in your open source PIV lab or something like that where you can use the cross correlation technique to get your flow inside the channel. So now we will see how cross-correlation works. So, as already discussed, like in PTB, we were tracking single particles; here, we do not do that exactly, but we track our group of particles and their integration windows. So basically, cross-correlation is finding how similar two different functions are. We will do some examples too, which will clarify things more.

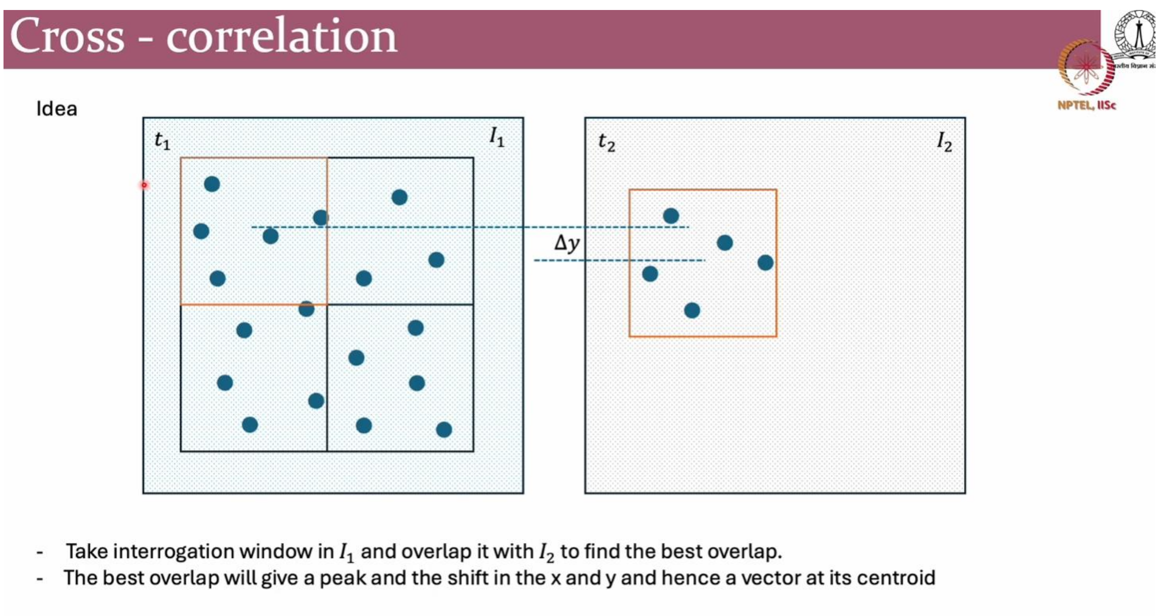
So suppose we have this channel and we have these particles, the cedar particles inside. These are the cedar particles, and we have this channel again. So what we do now is we divide this whole flow domain into a number of windows, which are actually known as the interrogation windows. These interrogation windows can be of any shape; they can be square, rectangular, or even circular, and the best part is you can change the size. As needed, I mean if you have a lot of particles, you can use a smaller window, and if you have fewer particles, then you can use a bigger window.

It actually depends on the requirements of the experiment that have to be judged very well. At the end of this topic, I will show you the effect of changing this interrogation window size. So, now suppose we have a group of particles inside this interrogation window, and then you actually do the cross-correlation for every window that has been created here. I mean, if it is like 4 x 8 here, then there are 32 windows, and for each of the windows, you will be doing the cross-correlation at different time instances. So basically, the idea is to track a pack of particle motion, not a single one, and then we estimate how much this interrogation window has actually shifted within the recording time Δ t, which we can set as needed.



So now let us see the idea in a bit more detail. So this is supposed to mean that you have an image that is I_1, and this is an image where you have some particles. Here, obviously, I have kept the particle density very low, but in actual cases, the PIV particles are very densely packed to get a better correlation and data. So now what we do is we have this image 1 at some time t_1, and then we divide this image into, say, 4 interrogation windows like this. Now we will just talk about this orange window.

Now we have another image I_2, which was taken at T_2 time; T_2 is basically $T_1 + \Delta T$. Now we see how much this orange window has shifted in ΔT time. And once, actually, what you can do is pick this image, pick this window, and try to find the best overlap in this image at T_2 time. So, basically what you are doing is picking one window, going to the T_2 time instance, and then trying to find the best overlap between these two. Once you get the best overlap, you actually get a correlation peak, the maximum correlation peak, and using that, you can say how much the particles have actually displaced.



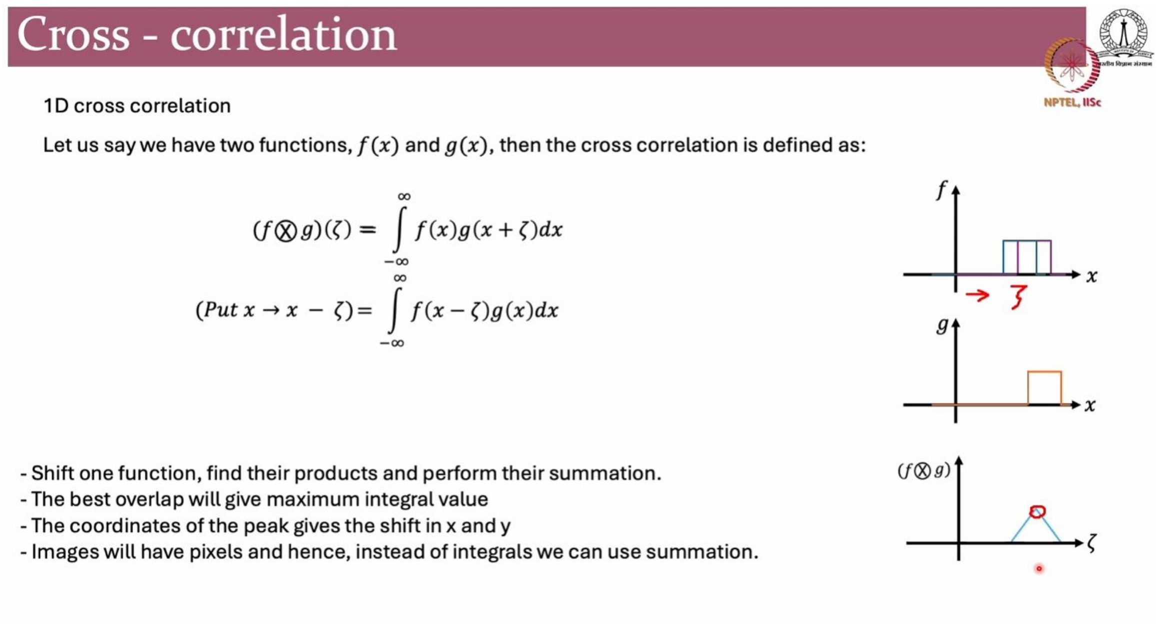
So, the basic idea is you take the interrogation window in I_1 and overlap it with I_2 to find the best overlap, and this best overlap, as I told you, will give a peak and the shift in X and Y. of this interrogation window. And if you get the x and y shifting values, then you can actually obtain a vector at the centroid of this interrogation window. So, if you are dividing this into four parts, you will basically get four vectors at the centroid of each of the interrogation windows. And in practice, cross-correlation shows how much the windows must be shifted to achieve the best overlap.

How much must your windows be shifted in Δt time to get the best overlap? So, what you want to find is the best overlap, and if you get the best overlap, you get a very significant peak, and that gives us the displacement of the windows. So, now we will see a simple 1-D correlation, cross-correlation basically, and how it works. And we will consider two functions, f (x) and f(y), and we will see how we can define the cross-correlation function for these two functions. So, say this is f (x), and you can see this is f (x), and it is a top hat function in this blue color, and we also define some g (x), which is something like this. To write the cross-correlation, we use this symbol.

So, basically, f x g should be a function of ζ, and it is actually defined as the integral from minus infinity over the whole domain of f(x) g (x) + ζ. And actually, since we are integrating in, say, x, we get a function in terms of ζ. And then, if you actually, as we talked about in the previous slides, are interested in shifting the interrogation window, when you shift it, you get the displacement in Δ x or Δ y. So, here, since we are just dealing with 1D cross-correlation, we will get some displacement in x. Now, let us displace this first function, and we see that if you displace this f(x) function by ζ, you can write something as the same thing, just that instead of f(x), you can write it as f(x) - ζ and g(x)dx.

And actually, when you do this, you get something like this. So, you can see a pink color function here, which is nothing but the f(x) being shifted. Basically, if I write it something like this. So, f(x) is shifted by some ζ value, and when you integrate these two things, you get a peak here. Now, if these two are not overlapping, then you would not get anything; I mean, the integral will be 0, and you do not get a peak that happens when, actually, your particles move out of the window size, which is very undesirable.

However, here you can see that this shift is basically the shift in the function, and this gives you the Δ x. So, what we did is basically shift one function and then find its products; say we have shifted this f(x) and we have found the product of this f(x) minus ζ and g(x), and then you perform the summation or the integral, and the best overlap will give the maximum integral value, and the coordinates of the peak give the shift in x and y. Since there is no y, we will just get how much it has shifted in x. And most of the time, we are dealing with images, so we will have pixels, and instead of integrals, we actually use the summation of the functions. We will talk about it on the next slide.



So, let us see how the same thing can actually be extended to a 2D cross-correlation where you can write something like that. If you have f and g cross-correlating, suppose ζ and I have taken another variable eta. So, you do a double integral over the whole domain, and you can write something like this: f(x), y, g(x) + ζ and y + dx dy. Now, let us shift. Now, if you shift in both x and y, then what we can write is f of (x minus eta) and (y minus eta) and g of (x and y), and this should actually give the cross-correlation and the shift in both x and y to get the best overlap.

Now, in terms of pixels, if we suppose we have m x n pixels in an image, then the same thing, the same formula can actually be written as this f and g cross correlating, and we can write it as summation signs instead of integrals, and you have this n equal to 0 from capital N - 1 and m = 0 -> m - 1. You write it as f (m,n), g(m+i,n+j). So, what we say is we are doing the summations over n and m, and we are shifting this image by i and j. Once you shift it, then you take the product of all these two functions, and then basically you are doing the summation over n and m. This gives you the cross-correlation peak, which is required to get the vector and the flow filter.

Cross - correlation

2D cross correlation

$$(f \otimes g)(\zeta, \eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)g(x+\zeta, y+\eta)dxdy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-\zeta, y-\eta)g(x,y)dxdy$$

Shift in both x and y to get best overlap

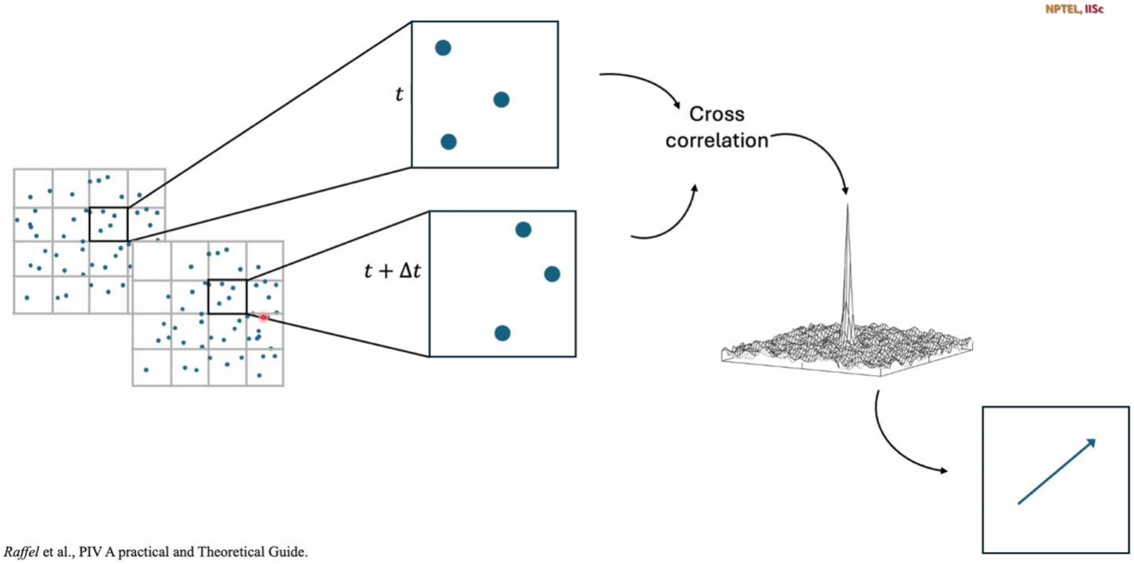
In terms of pixels,
Let us say we have M x N pixels, then

$$(f \otimes g) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f(m,n)g(m+i, n+j)$$

Here, we have summations over n and m and we shift one of the image by i and j followed by taking the product

So, cutting a very large story short, PIV, as already told, means that we are not actually tracking a single particle; we are actually seeing how a bunch of particles pass through a window, and basically, it is an Eulerian method, unlike the PTV, which was normally. Lagrangian method of tracking particles. So here we are doing nothing but searching for a zone in a future frame, basically at Δ T, that matches well with the current frame's interrogation window. So, to sum up, you have a flow field that you have actually divided into 4, 4 x 4 grids; you have particles, and you actually see that at some time t you have this pattern, and at some time t + Δ t, you have this one. Basically, I am just showing this for this interrogation window, but what PIV will do is actually try to cross-correlate each of these interrogation windows with each other by shifting and finding the best overlap.

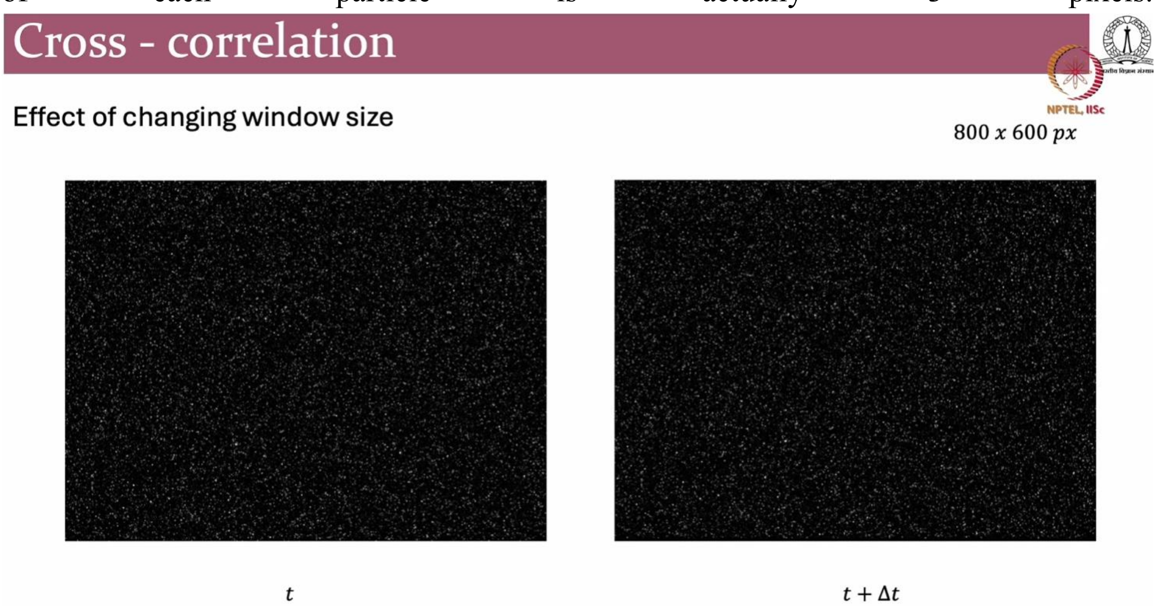
Cross - correlation



Raffel et al., PIV A practical and Theoretical Guide.

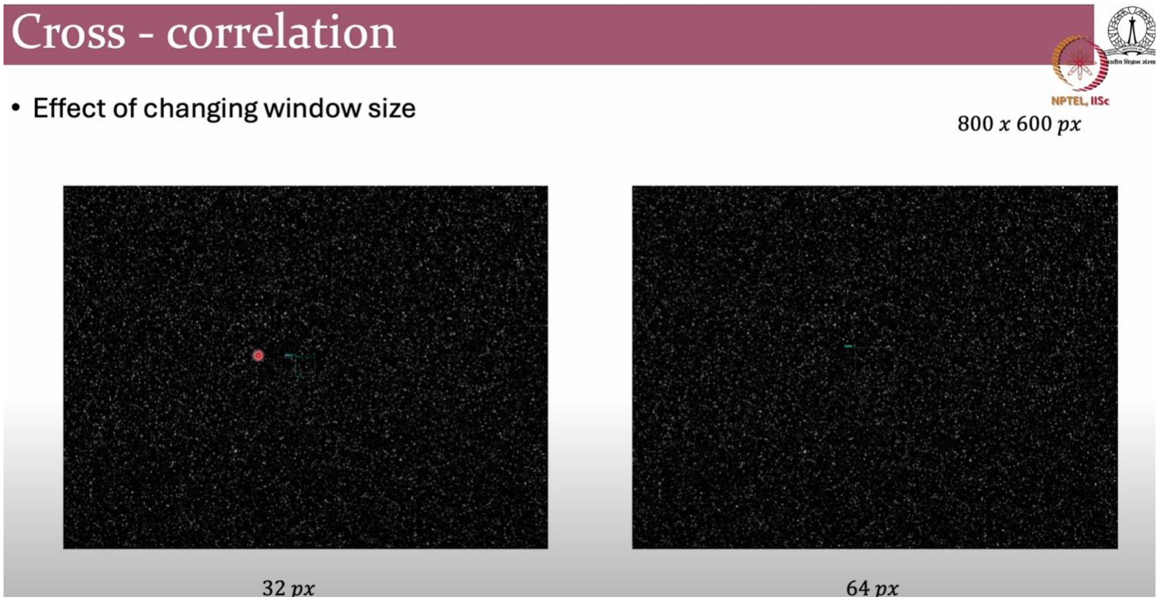
So, basically, this is the future frame at Δ t, and this is the current frame, and when you actually get these two images. You try to do the cross-correlation with these two images, and then you get a peak that looks something like this. This has been taken from Rafael et al. It is an excellent guide for anyone to understand PIV. And then, if you get this peak, you get the displacement, and then you get a vector for the interrogation window you are looking into.

So, as such, you will get this: if your flow field looks like this, then you should get around 16 vectors for each interrogation window you have created. So, as I told you, we will be seeing a very small comparison between two synthetic images to actually investigate how the vectors change if you change the interrogation window size. So what I have done is basically consider this synthetic image of a flow field. Which was actually generated using MATLAB; the latest version of MATLAB does this, and you can actually generate some synthetic images by providing the parameters. So, here basically the number of particles it is using to simulate this image is around 100,000, and the size of each particle is actually 3 pixels.



So, I took this image at some time t, and the total number of pixels is 800 by 600 pixels, and at some time t + Δ t, I generated this image. So, these are actually typical images, typical PIB images, and it looks like this. This is very well-lit, and you can see the particles very clearly. And the density, as you can see, is very densely packed, unlike in the case of PTV, where the particles normally have to be sparsely packed to track them correctly. So, yeah, and you can see now that just by looking at it, you would not be able to see the difference, right? I mean, because the Δ T is very small.

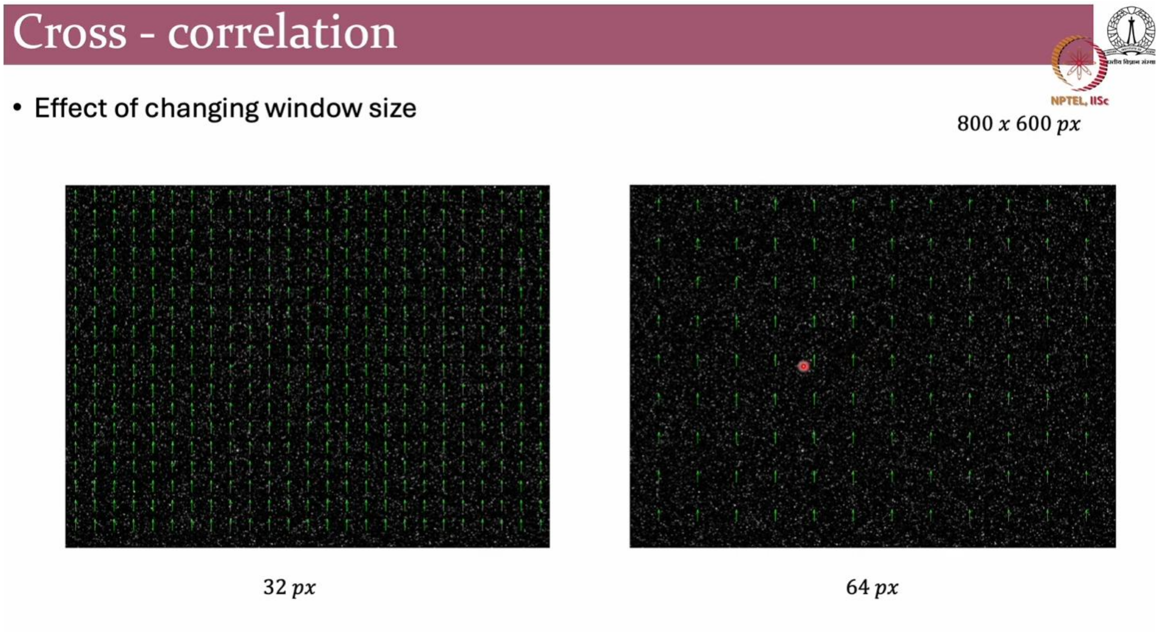
But if you play it in a pattern on a loop, you can see the particles shifting. That should be done. But now what we do is what I will show you. So here you can see a blue box, and this processing is done using PIV Lab in MATLAB. And by the way, the PIV lab is now also available as open-source software.



They recently released it so everyone can try this because it's a good hands on and PIV lab is very well used in the research area and it's a very well established software and very easy to use also. So, here you can see I have created one window size which is 32 pixels x 32 pixels; it is a square box. Here, I have created a box that is 64 x 64 pixels, and we will see when you actually do the cross-correlation and PIB between these two images how, basically, you will see the difference in the vectors between the previous two images. So for 32 x 32, basically, you get something like this. So, as I told you, I generated a synthetic image, so you shouldn't expect it to be like a parabolic profile, which our intuition normally suggests.

This was just a linear shift of the particles in the y direction. So, you can see it is giving a uniform shift almost everywhere. And when you look at 64 pixels, you can see the vectors are very sparsely aligned. I mean they are aligned, and they are very sparsely packed. So, this is what changes when you actually change the window size because you are dividing the whole image into much smaller grids now, and hence you get a lesser number of vectors, and actually it depends on your need as to how you want to present your data.

Also, it's very useful; here I have generated a synthetic image, so you can play with the image and get nice vectors, but in real life... In a scenario where you are doing exhumants, it is very important to take into account the particle density and the lighting, because normally lighting plays a very important role in PIV; we normally use lasers, which are not very necessary, but we use lasers for this, and the intensity and the depth of the laser and those factors should be well known beforehand, so that Your particles are very well lit, and you can actually get such good vectors without much noise. Right, so yeah, you can actually go much, much lower; you can go to 12 pixels and those kinds of things to get much more closely packed vectors.



But yeah, it will start giving erroneous results if the particle density is not very high, and if you keep on going to a very smaller size of window, you will start getting something erroneous that won't make any sense. Yeah, okay.