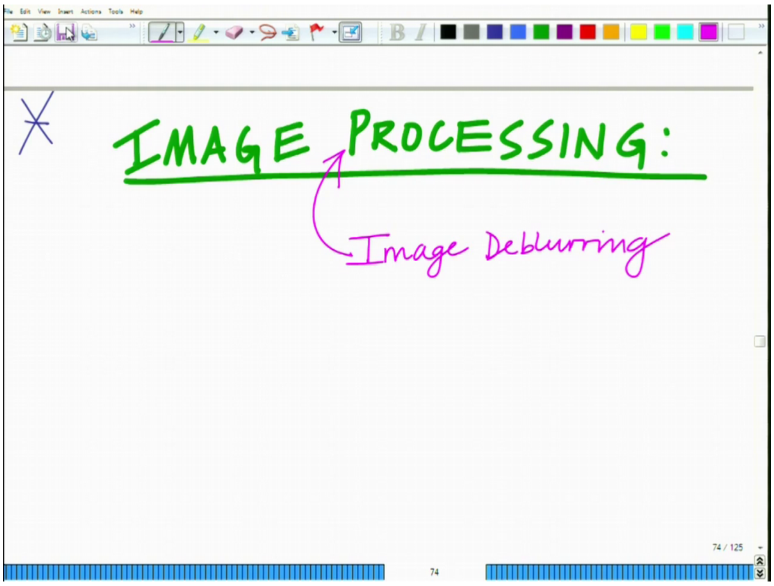
Applied Optimization for Wireless, Machine Learning, Big Data

Prof. Aditya K. Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture – 44
Practical Application: Image deblurring

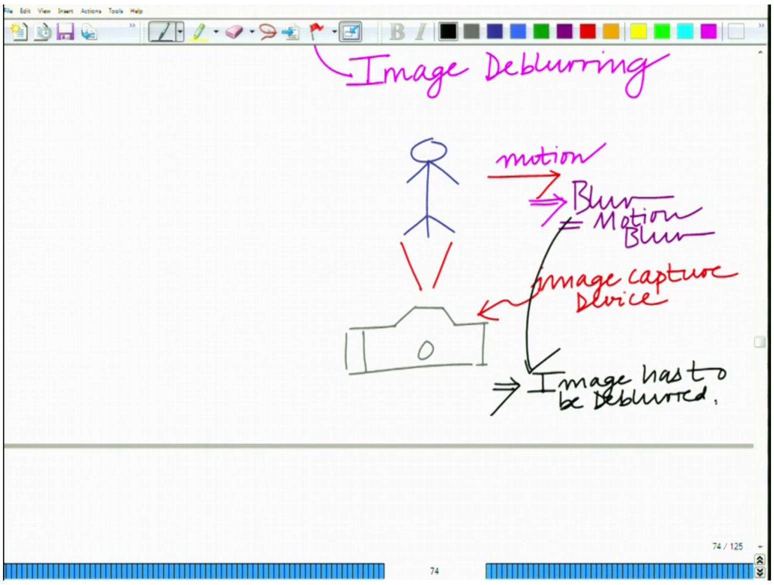
Hello welcome to another module in this massive open online course. We are looking at various applications of this paradigm of least squares that is the different applications where least squares can be employed. And in general least squares has, is a very flexible and powerful paradigm and is that can be applied in a variety of scenarios. In this module let us look at another interesting application and that will be in the context of image processing specifically in the context of Image deblurring ok.

(Refer Slide Time: 00:46)



So, you want to look at yet another interesting application and. In fact, there are tin number of applications of the least squares paradigm. So, just to illustrate the versatility of this framework, let us look at an application in the context of image processing specifically in the context of image deblurring ok. So, we want to look at the deblurring of image.

(Refer Slide Time: 01:24)

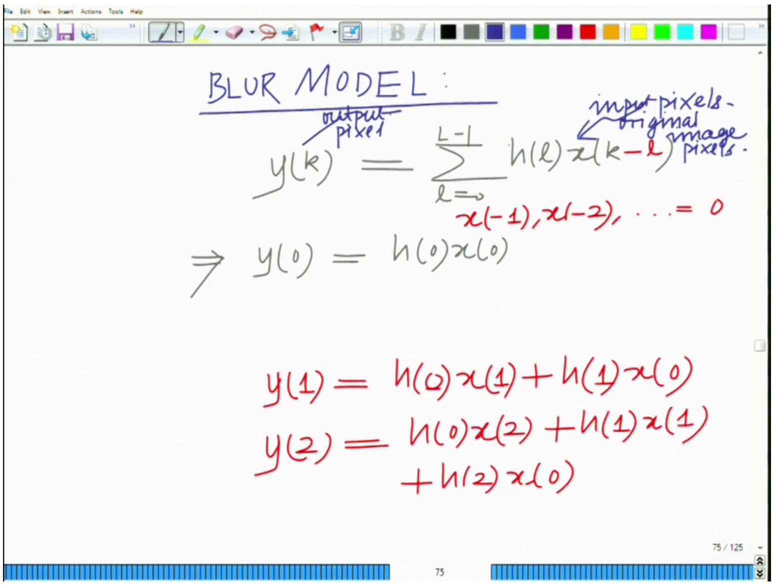


Now just before we formulate the mathematical problem let just look at what it the physical relevance of this; what happens is when you trying to capture the image let say you are trying to capture you are trying to make a figure of a person over here and when you try to capture an image of a person. And so, this is an impression of my camera and I am trying to capture the image of this person. And now, if this person or this object is in let say.

So, this is basically image captured device which is your camera or can be in like this can be extended to the context of videos also. So, this is not necessarily applicable only for images, but can be applied also for videos. If this object which you are trying to image or which whose image you are trying to capture is in motion then this leads to a blur alright. So, this remember when you have typically you might have seen. In fact, the blur effect that is artificially applied for instance on things typically on the images of let us say cars or vehicles that I have captured to give the impression that something is in motion that an object is in motion and when the object is in motion that naturally gives rise to blur.

So, the blur effect is basically associated with motion and blur can also it is basically degrading effect of an image and it can also arise from several other factors. So, just environmental factors, atmospheric factors, motion of the wind motion etcetera. So, in general motion leads to blur and this is termed as motion blur; motion is one of the predominant causes of blur and therefore, to get clean image implies image has to be deblurred. Now one way to model the motion blur is the following let us say you have the output pixel y of k.

(Refer Slide Time: 03:41)

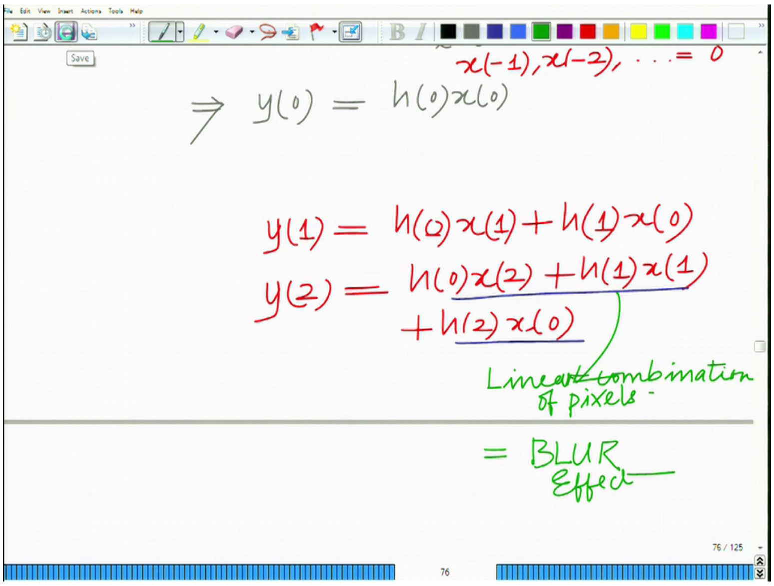


So, the model for blur model let us look at it. The blur model can be described as follows; let say you have an output pixel y of k that can be described as l equal to 0 to L minus 1. The input pixel or the blur kernel h of l times x of k plus l ok; so, let me just write this equation on for instance we have y of 0 equals h of 0 x of 0 plus h of 1 x of 1 plus h of L minus 1 x of L minus 1 ok. And you can also write this as y of 1 or you can just or you can make this that is fine it does not matter, you can write this as y of 1 equals h of 1 times x of 0 times x of 1.

Or let me just write this as follows I can write this is as h of 0 times x of 0 y of 1 is h of 0 times x 1 just to make this system causal or those it does not matter it does not really matter because image processing can be non causal. So, this is h of 0 times x of 1 plus h of 1 times x of 0 and I can write y of 2 equals h of 0 times x of 2 plus h of 1 times x of 1 plus h of 2 times x of 0, you can take the past pixels here to be 0. So, you can write this as h of l x of k minus l. And you can assume that the past pixels for instance x of minus 1 x of minus 2 so, on equals 0 that is this is a causal that is this is a signal which is 0 for n less than 0 alright.

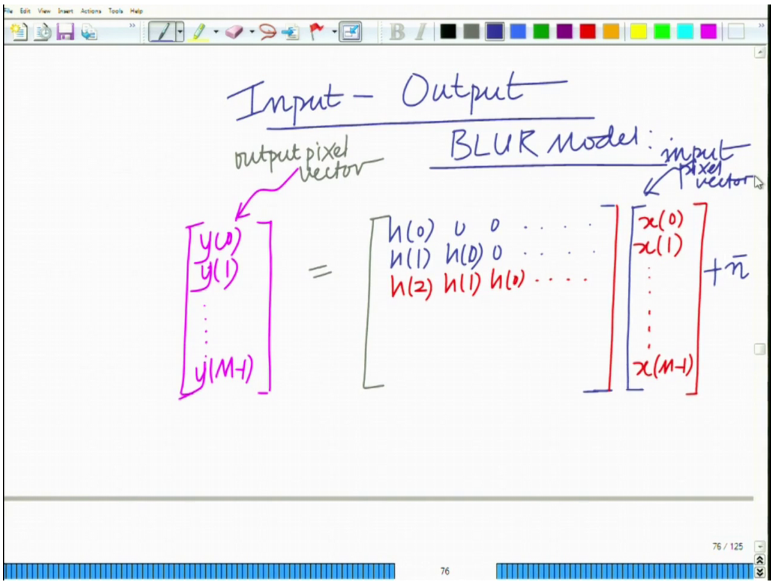
And so, what you can see here is that each so, this is the output pixel. So, this y k this is your output pixel and these are your input or original you can think of this as input pixels or original image pixels ok. These are the original image pixels and what you can see is that the each output pixel is a combination is a linear combination of several input pixels. So, you cannot you are not getting the crisp original pixels, but each pixel is sort of merged right, each pixels are off mashed or combined along with other pixels. And, that is what gives the blurring effect that is when you combine that is when you are not getting the clear individual pixels, but rather your getting a combination of these pixels that is x y of 2 the combination x of 2 x of 1 and x of 0. So, you combining these pixels aright and that is what gives the blur effect.

(Refer Slide Time: 07:17)



So, the linear combination of pixels is what is giving you the blur effect. The linear combination of pixels is giving you the blur effect.

(Refer Slide Time: 07:47)

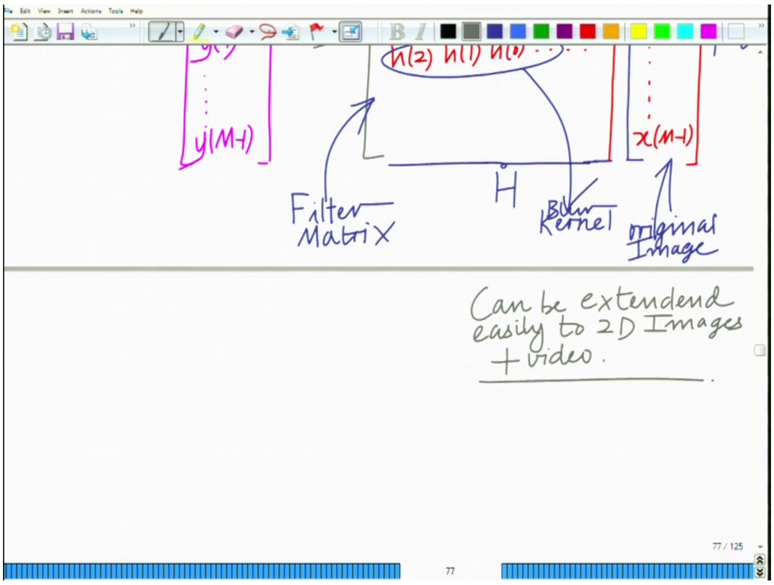


And therefore the input output blur model can be represented as follows let us say you have a group of output pixels that is which you are representing by the victor y of 0 y of 1 up to y of let us say M minus 1 that is total number of pixels is M. So, you call this your output pixel vector. We are considering a single of column of an image. So, this is your output pixel vector this is equal to well h of 0 h of 1 times h of 0 and so on and so forth.

And, this matrix has an interesting structure this is in fact, this is x of 0 x of 1 up to x of M minus 1 and this second row is h of 2 h of 1 h of 0 and of course, you can also have noise. Now, in addition you can also have noise, but let me just ignore this for a little bit just to simply; this model little bit although in technique in practice you can also have noise or let us make this. So, let us add the noise it does not matter ok. And now what you can see y of 0 is h of 0 times x of 0, y of 1 is h of 0 h of 0 times x of 1 plus h of 1 times x. So, it also depend so, it also combines both x 1 and x 2 x 0.

Similarly, y of 2 combines x 0 x 1 x 2. So, each pixel is a combination a linear combination of the pixel itself and the original pixel and some neighbouring pixel and that is what gives the blur effect. And this matrix which so, this is the output pixel this is the original input pixel vector we are considering as I said a single column of pixels.

(Refer Slide Time: 10:22)



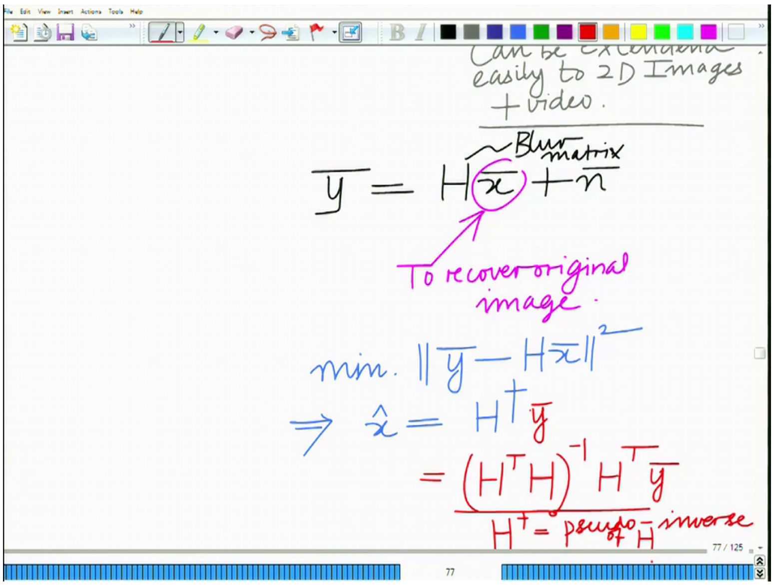
And this is basically your filter matrix, this is also known as a filter matrix. Because you are representing the blur by this filter this linear this filter represented by this linear transformation characterized by this matrix, I can call this matrix as matrix H. This is your filter matrix or your blur matrix. And this filter which you are repeating along the row this is also called the Kernel or this is basically your blur kernel alright. So, I can represent the blurring effect in the image as this linear system.

So, the blur model can be the blur effect. In fact, this can this model can also be introduced used to introduce blur alright to get the blur effect in images; for instance you want get the effect of an object being in motion such as a car being in motion this linear transformation can be used alright. So, this can use both ways either to recover the original image from the given output vector y bar or given input vector y bar x bar to introduce the blur effect one can use this linear input output system model. Alright, now the point that in the problem that we are considering is the other way not specially that is given a blur image how do deblur it alright.

And now once you formulate this problem. So, this is your original image I am representing a single column although this can be easily extended to a 2 dimensional original image and this can be easily extended to 2 dimensional. In fact, three dimensional images also which is nothing, but video that is your x axis y axis and in a time alright. So, one can have 3 dimensional blurring effect so, as to speak so, alright.

So, you can have original image. And this can be extended, I will just note here can be extended easily to 2 D images plus video and therefore, how to reconstruct the original we know that.

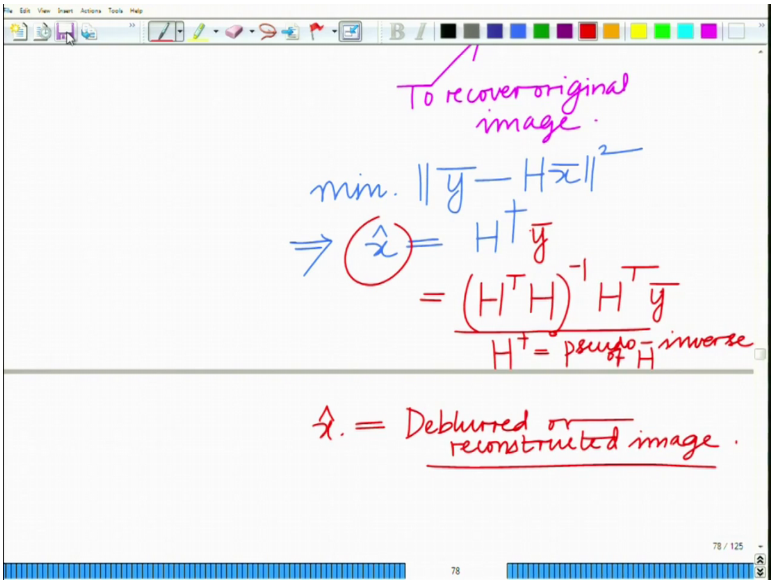
(Refer Slide Time: 12:26)



Now, we have this model output image y bar the equals the blur matrix H x bar plus the noise vector. So, this is your blur matrix and to reconstruct the original image or to recover we now apply the least squares alright. And therefore, what we do is minimize norm of y bar minus H x bar square. Implies the estimate or the reconstructed image or the deblurred image is the pseudo inverse of H.

We are not introduce this notations. So, far let me just describe this is nothing, but H transpose H inverse H transpose y bar where we are denoting this matrix by H dragger. This is known as H dragger equals the, we already said this is acts as a left inverse this is the pseudo inverse of H not the inverse, but the pseudo inverse. This is the left inverse of H.

(Refer Slide Time: 13:58)



And this x hat is now your deblurred or reconstructed deblurred or reconstructed image. This is your deblurred or reconstructed image and therefore, what you can see now is that yet another interesting applications of the least squares paradigm, is a very interesting application. It can be applied we already seen another application that is for channel estimation in a wireless multi antenna system.

It can also be used for deblurring of images in image processing. And therefore, reconstructing or recovering the original images alright. And therefore, the least square paradigm in general has many applications several applications; it is one that arises very frequently. In fact, in several different areas signal processing and communication and also other scientific disciplines alright. We stop here and continue in the subsequent modules.

Thank you very much.