Image Signal Processing

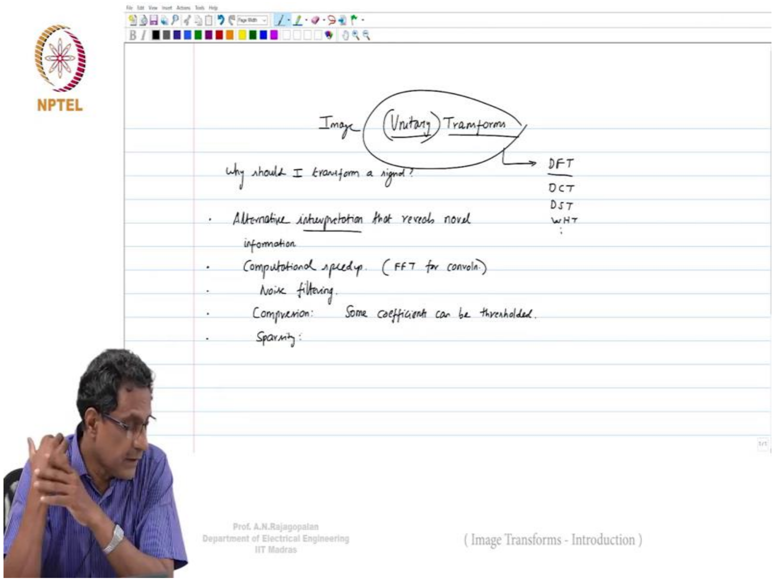
Professor. A. N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture No. 36

Image Transforms – Introduction

Today, we will start with a new topic, what are called unitary transforms?

(Refer Slide Time: 00:22)



So, now so it is a kind of shift between whatever we have been doing till now, so this is a complete it is a big shift now, here from let us say what what we did till now and this unitary transform business is about so for example, at one one special unitary transform is actually a DFT, which which is something that you have all done at some point or the other at least at least a 1D DFT have all done, 2D DFT I do not know how many of you have done it, but 1D DFT all of you have done.

Now, so for example there are kind of say several of them, DFT is just one, then there is something called DCT a cosine transform and there is something called a discrete sin transform, then there is Walsh Hadamard transform, there is so many of them that come under this instead of instead of taking let us say DFT and then DCT separately, one at a time, instead of that you could you could also you could also do it I mean up to up to how they teach. so ,the way I see it is, you know, I would rather kind of see create an umbrella of you know umbrella that talks about unitary transforms in general.

And then and then after after we have a framework like that then we would like to go and check out each one of these. So, in the sense the framework will be kind of common and then and then on each one of these will have some special properties I mean they will have an affinity for certain kinds of for certain kinds matrices and so on. So, for example DFT I told you, when we when we actually did that did that LSI, LTI, LTA system matrix, so at that time we said that you know it can diagonalize a circulant matrix.

Similarly, when DCT for example, if you look at Marcos, Marcos kind of a process, you can show that a DCT is more optimal, not optimal is superior to a DFT and so on. So, this association that sense each will have an affinity for certain classes of classes of kind of say data and either either you kind of see it from that point of view from in terms of data or you see it in terms of what it can do to a system matrix and so on.

So, so it is kind of a fairly wide thing, we will try to do what we can and the idea would be to sort of it, so it would be to sort of create a big umbrella under which each one of them can be just examined as special cases, which will make it easier. And then and there is a nice sort of a structure to all of this, we will be going to missing all that structure if I simply wrote down in the equation where DFT and said this is the 2D DFT.

Now instead of that you know if there was a natural way to get on to a 2D DFT from our 1D DFT that looks a lot more nice actually that structure exists, it is for us to explore it or not. So, I thought I thought what we will do is we will kind of go from what we already know in terms of 1D unitary transforms, then we will see how there is how there a natural vehicle that takes you from a 1D transform to actually a 2D transform, which will also set the stage that tomorrow if you have to do a 4D transform you will automatically know what you should be doing.

So, so once you know from 1D to 2D how it goes then 2D to 3D or 3D to 4D it depends upon what you want to do, but all of that we will look into straightforward then, look like it is just one step away, because that structure is all inbuilt, which is what which is you know which will be kind of say which will be sort stupid to miss. So, so we will kind of go like that, first of all, when you talk about actually right the real topic is image transforms rather put this in bracket, so basically it is about image transforms and mainly it is about it is about unitary transforms.

So, first of all when do you why do you think that people want to even go to a transform I mean why does one take some data and then do some transformation on that? So, why why study let us say, why should I why should I kind of transform data? Assume that whatever I (())(04:33) have any image or a 1D signal speech or whatever it does not matter, so why do you why should I transform an an image or transform an image or else transform a signal let us say let us not put 1D 2D and all, why should I why should I transform a signal? There must be different reasons, one reason, what is that?

Student: (())(04:54)

Professor: So, spectral information.

So, in a in a sense that is like an alternative interpretation, so you see the signal in one domain and then you see it in another domain and that seems to seem to reveal things that are not kind of say directly revealed in one domain. It is like an alternative interpretation, it is like an alternative interpretation that actually that reveals new information, new in the sense that reveals novel information, let us say, nothing new because whatever is there in the in one domain is exactly what is there in the other domain but something that makes it interpretable that makes it easier, so which is why I wrote, you know interpretation.

So, maybe your interpretability becomes kind of say better, it rather than, then what are the order the other reasons?

Student: The operations are (())(05:49)

Professor: The operations are?

Student: (())(05:51)

Professor: Exactly, so not only may be simpler but even faster maybe.

So, one could be a computational speed up you might want a computational speed up similar to the case that you know we have, so instead of doing a convolution in time domain images go use the FFT, compute the compute DC DFT and take the product and take the inverse that is faster, so computational speed up.

So, for example FFT for the convolution is one example that is okay and write down if you just even if we talk from terms of Fourier transform, that is fine. Then then maybe noise filtering or something, like for example, you could have you could have you could have a signal that is getting super imposed with some sinusoidal noise and so on like for example the 60 hertz fluctuation that we have.

These are all easily you are going of say determinable because once you go to the Fourier domain, you know that that it exists at that frequency and therefore you can have some kind of a filter, whether can remove it or even in general when you want to do when you want to do noise filtering what you are going to say, typically do you may want to go either either, you know, for example, if you know where that frequency is occurring it makes it easier something like this an electric supply fluctuations and all.

Sometimes it maybe just some other random kind of noise in which case also ready may want to do some filtering and perhaps the filtering becomes quite as a better if you go to some other domain, noise filtering then, what else anything else that you can think of? I have a few more things here, what about the cut compression? A compression in the sense that you go take let us say a Fourier transform and then you realize that some of those questions can be actually dropped in the sense and then you take the inverse you look at the image, so so it is like saying, depends on the application.

If you have an application where let us say you are extremely sensitive to what is there in the image maybe you cannot afford to drop too many things. But if you feel that only some certain information needs to go and that any way can be conveyed, even if I even if I chop off a few things right, then you can say what you said, so basic compression, compression is one thing of course this is not the same as doing a compressive sensing, there you only sense what you need, whereas in compression you sense a lot more and then and then you sort of sit back and worry about what can you throw.

So, ideally we should know how much to sense but then most of the time we do not know how much to sense. So, we just sense whatever we can and then we are going to worry about then we go to a transform domain then we will see what can be thrown off. Then anything else compression, so let us say let us say some some of the some of the insignificant in the sense that what is insignificant for an application, it does not mean that always you can do that.

So, so some some coefficients can be when I say coefficients those are in the say transform domain, some some coefficients can be can be kind of say threshold in the sense that you can simply remove that and and then get of say that the image is still reconstructed fairly well for my application.

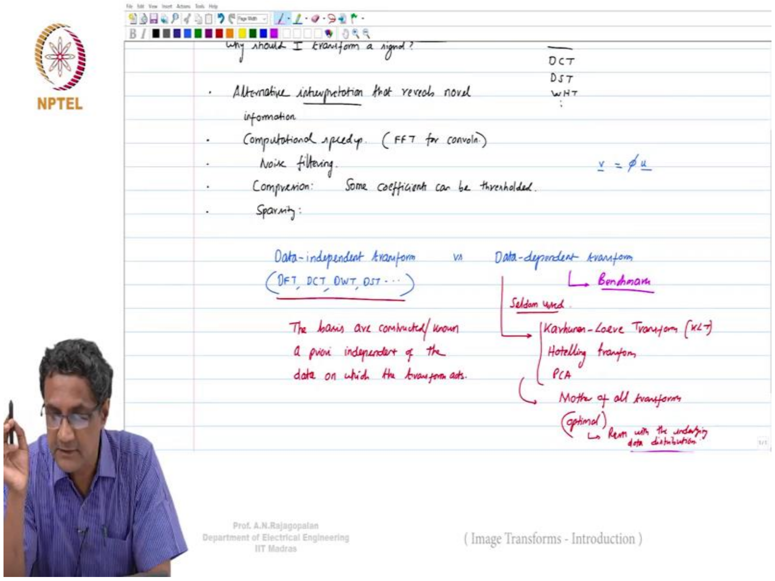
Then there is one more reason the, what is called sparsity? So, for example at sometimes you know if you look at the image sort of the representation, it might actually so happened that in one domain for example (())(09:20) let us say when you see the image, it looks like looks like there are so many coefficients everything looks important all these n square, n square values that you have on the grid, but then if you were to take a wavelet transform or something, then you will certainly find that it has a sparse representation there.

Sparse in the sense that a few coefficients sort of you know seem to tell the whole story and then which means that is a lot of energy gets packed in a in simply a few coefficients. Another example would be like to kind of go back to what we did yesterday, like we said on the one hand you have a space variant blurred image for which you can ask for a PSF at every location instead of that a sparse representation would be to go to the go to the kind of a say motion space, the pose space where you know that the camera has only gone through some 5 or 6 poses and then say that right have an (())(10:06) norm there and say that is all for the poses rather than solve for each of the blur kernels I mean is giving an analogy because that is something that we did only recently.

So, sparsity again means that you know and apparently natural images have the sparsity in some transform domain or the other wavelet is something mostly people find that natural images have a sparse representation, but then there are so many others contour like transform curve, so many transforms that have come up wherein certain for certain applications they will have a very nice kind of a sparse representation depends upon again your application.

So, so so the idea is that and of course, you know, then you know then there are then there are others or anyways, so the main things are these, you can of course add on a few more. Now, the now the way I would I would kind of you know I would like to whatever uncover a cover whatever you call that would be to kind of would be to look at it in terms of you know under under under kinds of 3 heads.

(Refer Slide Time: 11:13)



So, first of all, it is kind of look at what do we mean by the data independent transform versus because you know one for me to write those 3 heads, we have to first kind of look at what do we mean by data independent, independent transforms versus data transform versus a data-dependent transform.

So, for example, when you kind of take apply a Fourier transform on a on a sequence let us say suppose I have a sequence u, it just like you know which has a linked in, let us say u0 to u1 and minus 1, now when I when I kind of apply apply a Fourier transform on this, that is like equivalent applying a DFT matrix acting a DFT matrix on this in order to give me a v, v will be your this DFT coefficient which in general is complex.

Now, we do not really worry about what is u, when we apply phi, you give me any u, once I made up my mind that I want to use a Fourier transform I do not really worry what is this u, for example, if you think of it as an image, I mean for an image, you have to get of write it in a slightly sort of a different form, but just to kind of just just for the sake of the some motivation.

So, for example, if I knew that it was a face, I do not change the basis or for example, if I knew that it was a it was a class of tree, so this image is coming a priori is suppose a new that it is actually a tree, I do not I do not actually I do not actually use that knowledge at all, I simply say it is going to be a it is going to be still the same Fourier basis its standard.

You do not change the basis you do not care what is input signal that is coming in. In a way it is good because you do not have to worry every time as to where is the signal coming from, what is its origin? Whatever be it, we simply have this kind of data sort of an independent transform, which is like a DFT or if you feel you want to use a DCT, but on all of those the basis that you choose is completely independent of the signal, you do not worry about what the signal is.

In some cases you can show that if the signal happens to come from a certain class then maybe this transform that you are using even though it looks like it is data independent, but then it could turn out to be optimal for that class, this is something that we will look along the way. But normally normally we do not really worry about where that where that sequence is coming from we simply apply.

So, so all your DFT is DCT is they all come under this, DFT DCT, and all of this we do not look at the data and sort of and sort of make up the basis. We do not construct, construct the basis by looking at the data, we construct the basis independent of all independent of the data, so DFT DCT DWT, whatever you know DST you name it, all these, these are all these all come under what we would call as data independent transform

So, so it has its advantages because because completely independent of data, so so so the bases are known a priori, so for example, if I know that we know that I am going to use a Fourier transform then the other guy, I do not even have to send him this basis because he knows that if you are saying the DFT then it means that he knows what is the basis that you are going to use. So, the bases are bases are a kind of constructed or known a priori constructed known whatever you want to call it a priori, independent of independent of the data on which the human transform acts, independent of the data on which the transform acts.

So, in a sense, it is good and as we know lot of the things that are happening are all with with respect to independent transforms only, very rarely do we do a data sort of a dependent transform. But you might ask you then, why why does it even exist? Why should you have some like a data dependent transform? The point is data dependent transforms these are seldom used in reality, so for example, because every time if you have to construct a basis and it is going to change it depending upon class of data is coming in than it means that say other end that guy has to know what it is.

And again that you might want to add few more examples to the data in this case that basis will again change, that meant for example, if you had the you know whatever 1 lakh images today or that data tomorrow you all will say 2 lakh images you might say I want to get a set of define the bases now. So, all of this is like an is like an incremental process wherein you can keep on changing the basis and then then then it makes it harder because then that the other end you had to you had keep kind of say updating the bases and so on.

Therefore, this is seldom, seldom kind of used, cannot that, I do not mean that right nobody uses that I mean again that I am going to show you applications wherever you know data dependent transforms play an important role, to saying seldom used in the sense that if you if you are going to say looking at transmission of these coefficients and all seldom does one use a data sort of a dependent transform, seldom used in that sense. But then what are their what are their main uses?

As been that so for example, there is one transform which is actually a data dependent transform, which comes under the name of KLT, what is called a Karhunan loeve, so it is like Kerhunan leove transform. So, what is called the what is called the KLT, this also comes under a different names, it is called the Hotelling transform, it is called the PCA some people call it PCA, so all of this means the same thing Holtelling, KLT Karhunan leove, PCA all mean the same thing.

Now, this transform, mother of all transforms in the sense that then you might wonder, what is that? So, so so actually what it means is that, that this transform, it is kind of optimal in some sense and and the whole point is that when you say that something is optimal, then that optimality has to be with respect to some data, it is not like so this optimality notion right rest with so those optimality, so, this kind of rest with the underlying rests, this notion rests with the underlying data, data distribution.

So, now the point is this, so so so what this means is that when you are applying a DFT or a DCT or for example some other transform which is actually data independent, you might might you might want to kind of ask is my transform optimal? Generally, generally you do not ask because you know, first of all you do not even know what does it mean to say that something is optimal. but then this guy in which is sitting there the KLT kind of you know can be looked upon as kind of a benchmark.

Even though even though right here you may never use the KLT, but then what it actually means is that right you have the KLT sitting right there at the top, being optimal for that kind of data, again dissolve data specific, when you say optimal it is it is for that kind of data, which is why which is why I said with this for a for a distribution.

So, for example, it could be for faces, what could be the ideal kind of kind of a bases that you should be using or for example, if you look at natural scenes again, so what kind of an ideal base should you be using? So, in that sense it is sitting right there at the top and even though you know that using that itself is going to even though you know, it may be optimal and whatever sense but then using that is not easy.

So, instead of that what you say is now for this data if I use this DFT or DCT or whatever, can at least check how close do I how close how close am I to this guy, because if I am close, if I have 3 or 4 of these data independent guys have and then some say and if I can if I can show that one of them is actually closer, then it would mean that you rather use that than could have use some of the data independent transform. Because it already gives you an idea that you are kind of close, it may not be the KLT but then it is kind of close the gate and and then there can be situations where a data independent transform becomes the KLT.

So, so in that sense there is also a bridge (())(19:29) crossover you may think that you know, you may you may not know but then you can actually you can actually, find out determine that in some cases, you know in some special cases what might happen is something that looks like data independent becomes the KL transform for a certain type of data. Again, it is not like KL transform revenue talk about KL transform that is again a data specific.

So, for example, you might want to look at a covariance of a certain structure as an example, so you might want to say that for that kind of a covariance matrix see for example, you must have done a Marcos process and all, so so so those guys have a structure and we know that most natural scenes follow that kind of structure, it may not be exact, nobody is saying that, everything strictly Marcos down, but we know that you know, something would depend on its neighbours and therefore we have a certain sort of a sort of correlation.

Now, when you have such a such a matrix, which actually encapsulates what happens in a real data in a sort of a statistical sense, it does not really it is more in a statistical sense, so in that statistical sense what you can now sort of see is that, if you try to if you try to employ for that kind of a data distribution, if you try to use a DCT, for example, you can show that a DCT can come closer to a KLT, in a sense that you will have a basis for actually DCT, you can actually construct a basis corresponding to this data and you can show that which is supposed to be the KLT basis and you can show that these bases look very similar, you know they may not be exact but you can see that you know, this is guy is looking very very close to that.

So, so which means that you get a sense for how close you are kind of, so among these transforms even though these are all data independent, but then some of them could be could be could be, very good in terms of being used versus someone else, so how do you pick? So, in that sense data sort of a dependent transform is something that we used to do the benchmarking, that even though this by itself you may not use for transmitting or anything, but then it serves us as a nice benchmark.

And then this other thing is because because it is based upon the statistical nature of data, it is also it is also used for many filtering application and all where you learn collectively, what kind of statistics or data has and therefore it even this noise filtering and all that we are going to talk about you can actually do it in a sort of a statistical sense, what are called Eigen filters and so on. So, in that sense I am just saying when I say seldom used I mean for transmission and such (())(21:49) and such cases, but then, but then it does find application sales value.

So, but our but our goal I will also get to say I will also try to see if I can you know, throw up a few applications of the KLT or the PC or whatever. Of course, I am sure you have all heard about the Eigen faces that came up like 20 years ago, I did not go too far that is that is a different issue. But then that was that was the first time when somebody said that somebody did face recognition without using the geometrical features of the face, but rather something which is really which is actually statistically derived.

And those eigenvectors they could show that they capture whatever it different features of the face and the linear combination of those could kind of help you reconstruct reconstruct and then do face recognition and so on. So, in that sense it has been around and you know and it still is there and again a different in fact I will show an application where let us say nothing else works, nothing else works, nothing else works in the sense that whatever we try, I will show an application where only a KL transform works, because it only understands what kind of data is there, none of your other techniques will work there.

So, again I am saying so this guy is this guy is kind of say special, so so basically we should not we should not we should not underestimated it just because we do not end up using it does not mean it is not it is not strong enough.