Real-Time Digital Signal Processing

Prof. Rathna G. N.
Department Of Electrical Engineering
Indian Institute Of Science, Bengaluru

Module No # 11
Lecture No # 53
Discrete Cosine Transformation

Welcome back to real-time digital signal processing lab. So, today we will see a DCT so how we are going to implement discrete cosine transform with quantization so, how much we can see the view our image and then look at it.

(Video Starts: 00:40)

So, the first will see the MatLab so you will be seeing that these are the students who have developed it as you can see this shows as an assignment 7 basically. Here we will be considering the histogram equalization that is when we feel that the image is little black so we want to do the histogram equalization to give the perception a little bit clear. The other one is the noise what it has so by using the gamma correction it is going to improve upon it.

And the other one is whatever demo I will be showing is with respect to DCT so we will run this code actually and then see how we are going to get the images out. So we will go to this folder. So, you will be seeing this is a GUI developed so as you can see the thing it has DCT histogram equalization and gamma correction. So we can load the image as an input so we will see that load an image so the first what we will do is load our first image as a baby image.

So, you will be seeing this is the image what it has got loaded. So, we will do the DCT so you will be seeing that the knob can vary from here to here, only the thing is in this case the slider values have not been given in steps of you have to go back to the code and see how what will be the steps of this slider is going to move. So, now we will run the DCT in this is the compression whatever it is shown with the slider so we will run it. So, this compression has not much change the output.

So, you will be seeing the histogram of the original image and histogram of the compressed image so almost they look alike. So, now what we will do is we will increase the slider little bit and then I will be doing the DCT. So, now what you have seen is only the complete thing has gone to black so the compression what we have taken is almost more than 70% what we can say a little bit reduce and then do the DCT of it.

So, we have got back the thing so this way you will know that what how much compression you can achieve with respect to this. Now if I want to do the histogram equalization. So, we will load a different image to see that whether we are going to look at that difference or not. So, we load the Einstein image so you will be seeing this so I can do the DCT or histogram equalization so we will run the histogram equalization.

So, you have seen that the original histogram is centered between this thing what is it black and then some of the grey values here and then little part of it are here and after that we have 0. But I want to have the uniform distribution of the thing so we do the equalization so you can see how the image has got affected. So, we can load the other image and then see whether it is going to give us better histogram equalization.

So, we will run the histogram equalization for this so you will be seeing that whatever that the background which were a little darker now you will be seeing that they have been exposed so by doing the histogram equalization. So, if you want to see background properly then you can do the histogram equalization and then run it. Now we will see how the gamma correction is going to happen.

So, now this is what the gamma correction so if I increase the gamma part of it and then we will run the gamma correction so you will be seeing that what happened is that you will not be seeing not much output. So, by this thing elimination of a lot of noise what happened is we have lost the thing so depends on what the gamma value is going to be. So here you are seeing that it is little darker at the background and then this is little bit prominent what you will be seeing this is how your gamma correction is going to happen.

And these are the histogram how when you do the gamma correction so this way you can play around and then see what will be your output you can get the thing with different images basically. So, now we will see the other way of doing it this is from the same badge one more student how he has implemented we will look at it. So, we will run the code again so it is from a different flow for folders we will be changing it.

And you will be seeing that this is the GUI what it has got here also you have to select an image file. So these are the images are stored in a different file so you will be seeing different images basically from black and white and color images what you have it. So, we will see that you know the personality of this person so we will do the DCT compression. Once again because the aspect ratio is more you are unable to see the figure here I will change the thing for the viewing thing and then come back from the settings.

So, we will change it to 100% and then we will run this code so now I will be aligning the thing. So, we will select the image. So, the images what you have seeing the thing this is the input image I am going to do first DCT compression. See here as you can see the compression factor is varying between 0 to 1 in this case so if I change it so we will do what is the compression factor? What we shown in the class that it has taken it as 4.8832 is the compression so do the DCT compression.

So you are not seeing much changes in the thing again I will increase the thing do the DCT compression. So, you are not identifying much of it so in this case it is between 0 and 1 what they have selected so this is the output was taking little time to come. So even with 20 compression factors so you will be seeing not much difference between the this is the input image this is the output image.

So, I can reduce it will do the DCT compression not much difference now we will see that what is the thing is going to happen will load a different image I can go to the images. So you will see that this image basically how is going to behave so we will do the gamma correction for it. Now it has loaded the thing so you are seeing that the plane is not much visible so we will do the gamma correction.

So, now as you can see that the background has also become a little brighter and then even the plane is completely visible and then you are seeing the whatever luggage carriers what you know of actually which are there and even the whatever the petrol fueling thing is also visible. So, I can do that so if I want to do the compression I can look at the compression how it is going to look like with the original image.

So, not much to the original image so you will be seeing that not much changes has happened so the in this case gamma correction has helped us to get more clearer one. So, we will see by doing the histogram equalization we will select an image file again, to show that their functions are working properly. Since it is running in loop it will take little time to come read the thing so this will see that this is a low light this thing so we will use this still it has not loaded the image.

So, you are seeing it this has a low lighting conditions as you can see we will do the histogram equalization for it still it is a processing. So now you can see that whatever the lighter so you are seeing the reflection completely in this thing water in front of it and this is become much brighter. So, this way you can have a different ways of doing your DCT compression and then your gamma correction and then histogram equalization by making a different image processing applications you can incorporate.

So, in this as an application what I will do is our audio file so I said we have not covered what I will call it as interpolation and decimation. So, our audio application one can use it at different sampling frequencies. In this case we show that input is 44.1 kilohertz input how we can convert it into 48 kilohertz is this demo is going to show you. So this way you can incorporate and what are the filters we will be choosing it is going to be FIR filter what it has been chosen of different length and then we will have 3 stages in this case to run the algorithm. So one is decimation, the other one is interpolation what we are going to have it in the second stage, and the last stage is going to do again decimation. So, to get our 48 kilohertz so you may not feel the audio file but from looking at their responses we will see that how it is going to look like.

So, you are seeing that how the changes is going to happen that is 3 stage output what you are hearing the sine wave from 44 kilohertz to 48 kilohertz which got modified. So did you notice any differences between the 2 so that is what the thing is going to be. So if you want to expand here so we will see that so you can do this zoom in to see the frequencies what it has got. So, from this is input that is 44.1 kilohertz to 48 kilohertz what it has got transmitted. How it 3 stages what it has done the thing that you will be seeing in this figure.

So, the original you will be seeing it in red and then what is it you will be seeing it is getting pushed to a little bit black and then little green and then the final is your blue. So, you will be seeing here this is original this is stage 1 what it has got little bit shifted and then later on this and later on this is the final. So, instead of because it is not possible to shift from 44.1 kilohertz it is an odd number to 48 kilohertz

So we do what we call it as multiple times so that the output will be visible that is what the MatLab what is shown the same experiment is available even in code composer studio. So, we will take up now the code composer studio. So, we will start our code composer studio what is happening as you are seeing it so it is creating it is little bit crashed I think so when shutting down so we will open a new this thing.

I will call it as V9 because recently I have closed it so we will launch a new workspace. All the files whatever demos have been shown so I have put it in CCS9 as you can see the thing so I have to pull out these files now into the new workspace and then I have to load them let it come up. So, we were discussing about the DCT and then I have the other and we will see little on echo today on the DSP processor.

Some of the equalizer and other things are giving little error in the thing so till I settle solve the problem once it is done in the next class we will be taking the demo of it. So, what I will do is I will copy these directories to here hopefully one shot it should work let us see if everything is copied otherwise we may have to copy the include files and then libraries. So, if it asks you to install you can in sometimes it gives an error so it is better to not install the new one whatever you have installed the original let it be there.

So, we will open the thing we will open from the project from the directory so we will put the thing I will be opening this folders hopefully. If you want to create a new project you can go and then create it since all that already these projects have been running. So, what I will be doing is you will be getting started and everything in your code composer studio so you can have a look at them.

So we will first run the DCT because I had to because it is a new workspace I have to do compilation first. And this is the C file in the meantime let the compilation happen we will see what is it. So, this is what you are hopefully you are able to read the thing I am reducing this also.

So, this is the code what we have it so here you are defining 64 by 64 image so you will be considering 64 by 8 to the power of 2 what you will be looking at it.

Because you are going to give image file dot h file has to be input as this or you have to read from the file as you know in the hardware it is going to be in a continuous memory what you will be storing it. We may not be able to read 8 by8 images correctly so you will be storing it as dot h file so if you see that image file dot h. You will be looking at it is a continuous so you will be seeing that it is a 64 by 64 the image has been taken from the MatLab.

And then dot h file values have been created and then you will be storing it in the continuous format like this which you will be in putting it into your CCS. So, this thing is going to run in what we will call it as even simulator you can run the thing. So, if you want you can have the Lena dot h also what you can look at it so and then it can be image file up to 128 by 128 what it has supported.

Otherwise we have the input directly from the camera you can feed it in and then do the compression and then take it out also. So, both DCT and IDCT have been incorporated in this code basically and the compression factor what is this factor array compression ratio is chosen as 4 here. So, if you want you can change the compression ratio here and then you will be seeing that whatever chromium and other things is 8 what you have taken it

And you will be allocating the memory and then you will be performing the DCT using this equation. As you can see that it is direct DCT has been implemented so less than n so you will be seeing that there will be of 4 for loops. Here so that is why order of n power 4 it is going to be so if you want to reduce it so you can go and then use a butterfly structure to implement the thing. So, you will be seeing that calculating you are doing the compression here whatever C R value what you have chosen the thing

And then so you will be doing row wise and then column wise. And then once you have stored this you have made the rest of them zeros whatever is less than what you are selecting the thing the values DCT matrix you are making them zeros. And then you will be storing it and then do the IDCT, so this is what the output is going to be. So matrix u comma v will be your IDCT. So, you will be seeing from this thing a main function is here so you will be seeing that total by compression ratio what you have said the thing.

And then compression ratio parameter what you have set it here square root of C R and you will be sealing that basically. So, not to have we will call it as decimal values are avoided only integer what you will be selecting the thing to the higher what you have sealed the thing. So, you will be seeing that some of the assignment so you are putting it as a test block. So your input image block wise what you are operating 8 by 8 blocks i, j what you will be working on.

This is the DCT you are passing the test DCT image and then block whatever the you have selected test block and then you will be giving x and y dimensions. Then once you have done that DCT which uses the compression also then i you will be doing IDCT as you can see there are so first we have to make clean and then run the project. As I have mentioned in the first class we have to go to properties so when we have this the latest version support elf file format.

But what we want is the legacy call file generation in the new one so I have to apply and then close hopefully we will not get any errors let us see the thing. That is the reason why it was cribbing that no this thing what is it make option is available. When you get this error it is better to go for sometimes even the warnings you may have to see that if it has to be eliminated you have to eliminate warnings.

So, in this case what it says is declared in the thing and then it is not used so that is fine so what we will do is we will build the thing. Because I have started the new project thing so what I have is I do not have the target configuration so I have to specify which board I will be using it. So, it is going to load the target configuration and it will ask me which one you want to have it I have to wait for it to give me that.

The first time it is going to take little longer time usually whatever demos all these days I was doing it so it was pre-built so that I need not have to waste time on the thing. Today because one of the workspace it is having a problem so I have to recreate it so it is running for the first time it is going to take a little time on the thing. It is asking me to update it so we will not do any updation here.

So, when we rerun the thing second time it is going to be much faster, one more thing what I can do is I can stop the debugging and then go back and then give it and I will call it as new. Because it was asking me for the target configuration file I will give it as configuration so we will try to finish it. So it will ask me which one I am going to use it here so what I have is as I told you in the beginning so I have a XDS 110 debug USB debug what I have connected here.

And then it is going to be LCDK6748 board what I am using it so this is the target configuration what I will be giving it and then I will save this one. And then I can do the test the connection whether it is getting connected or not so you will be looking at it software log files and other things so it is connected to the board or not what it is a testing. Some of this scan test what it says is it is failed anyway finally it says scan has succeeded.

So that is how the board whatever you are going to connect it you can look at them so we will recompile the thing. So, and then see whether it will go it is as you can see the debugger is going and loading on to the board. Some of whatever you have configured your memory you can you have assigned calloc basically. So, you can release them later on you will be freeing the memory once you come out of the loop.

As you can see you will be freeing them in the end, all the blocks what you have assigned the thing it they are going to be freed. So, now you will be seeing that everything has got loaded onto the board and your pointer is at the main here. So now you have got the icon to run and then if you want to break you can do the thing. So what we will do is we will put a break point here so that and run the code.

So, it has finished it so you will be seeing that whatever the modification values are dimension of y x and then i and j k; k has gone up to 64. And then what we will see is because it is in this case although image analyzer is there it is giving little problem in this version so what we have to do is we have to view the memory basically. So, whether it has got so we will go to the memory browser and then I can give output image is the where it is getting stored as you can see coming from the test IDCT the value is going to be stored here.

So, you will be seeing that it is a floating point number so you will be seeing that here it is stored in your data memory basically 1d what it shows and this is output image all the values what it has stored. So if you want you can write this memory back into a file and then you have to read it properly so this is a floating point number what it has stored. So, you can see that the values have come so some of the values if you want to see with respect to the original value so you will be seeing that reconstructed will look at it.

It is 4 1 0 0 what we have it so I can give it as input file we can dot h file what we can open some values what you will be seeing is 1 0 0 or whatever so we have to check our memory. So, once we close the thing the output file is not going to be available so what you have to do is you can copy it and then paste it and then look at it. So, the other one we will see here also there will be a problem with the thing because this is from the book of you will be converting from Y C b C r to R G B.

So, this is the DCT test dot c what it is gives from the co book what it has been selected so I have to have the here also I have to call my cc xml. So, what I can do is I can copy this one to here so that I need not have to again put the thing and we have to see its properties again you will here it is chosen as legacy COFF file here so I need not have to worry. And then you will be when you are doing the thing the family I will be selecting it as C6000 and you will be seeing it as LCDK6748.

And then the compiler version for this board is although there are higher versions of it is better to use a tib 7.4 in this case. So, we will just do that I am directly gone to do the debugging because it is asking me there is an error. Here what happened was the scope gives 55 version I have this one DCT has been written in assembly. So, assembly with respect; to 6, 7 and then 5, 5 are different rest of this as you know now why assembly coding is particular to one of the board.

So we will modify this DCT to what we will call it as to the 6, 7 processor. And then we will see in the next class how we are going to implement this using assembly programming little bit of it what I will give a little flavor only multiplication what we can do it and then we can look at the rest of it. If you are interested in assembly programming you can learn it and then implement it thank you.

(Video Ends: 37:44)