Taylor Larrechea

Dr. Gustafson

Math 362 Fourier Analysis

October 20, 2017

Class Prep 10

Section 4.6

Key Concepts: In this section we develop the DCT II, the discrete counterpart to the CT II. We then discuss DCT II block thresholding and apply it to a voice wave, where we will see improved thresholding results when compared with block and regular FFT thresholding.

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> LinearSymPlot(2,-1) |  |
| Input Commands | Output (Plot if Applicable) |
| >> x=[1,2,3,4]';  >> EvenSymPlot(x)  y =  1 2 3 4 4 3 2 1 |  |
| Input Commands | Output (Plot if Applicable) |
| >> EvenSawPlot(8) |  |
| Input Commands | Output (Plot if Applicable) |
| >> x=[1,2,3,4]'  x =  1  2  3  4  >> DCTII(x)  c2 =  5.0000  -2.2304  -0.0000  -0.1585 |  |

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> x=[1,2,3,4]'  x =  1  2  3  4  >> DCTIImatrix(x)  c2 =  5.0000  -2.2304  -0.0000  -0.1585  x2 =  1.0000  2.0000  3.0000  4.0000 |  |
| Input Commands | Output (Plot if Applicable) |
| >> x=[1,2,3,4]'  x =  1  2  3  4  >> DCTIImatrix(x)  c2 =  5.0000  -2.2304  -0.0000  -0.1585  x2 =  1.0000  2.0000  3.0000  4.0000 |  |
| Input Commands | Output (Plot if Applicable) |
| >> x=[1,2,3,4]';  >> N=length(x);  >> y=[0,1,0,2,0,3,0,4,0,4,0,3,0,2,0,1]';  >> z=0.5\*sqrt(2/N)\*real(fft(y));  >> z(1)=z(1)/sqrt(2);  >> c2=z(1:N) | c2 =  5.0000  -2.2304  0  -0.1585 |
| Input Commands | Output (Plot if Applicable) |
| >> x=[1,2,3,4]';  >> DCTIIfft(x)  c2fft =  5.0000  -2.2304  0  -0.1585 |  |
| Input Commands | Output (Plot if Applicable) |
| >> [z,sr]=audioread('FourierAnalysisIntro.wav');  >> DCTIIsound(z,sr,256,95,6000)  Percent\_Reduction =  'The percent reduction is 94.921875.'  Compression\_Ratio =  'The compression ratio is 256 to 13, or 19.692308 to 1.' |  |
| Input Commands | Output (Plot if Applicable) |
| >> FFTthreshSpec(z,sr,256,95,6000)  Percent\_Reduction =  'The percent reduction is 94.921875.'  Compression\_Ratio =  'The compression ratio is 256 to 13, or 19.692308 to 1.' |  |
| Input Commands | Output (Plot if Applicable) |
| >> [z,sr]=audioread('FourierAnalysisIntro.wav');  >> FFTsoundthresh(z,sr,95,0.5,0.54,60,6000)  Dominant\_frequency =  'The dominant frequency is 117 Hz.'  Percent\_Reduction =  'The percent reduction is 95.000374.'  Compression\_Ratio =  'The compression ratio is 80286 to 4014, or 20.001495 to 1.'  Run\_Time =  'The run time was 0 minutes and 2.676745e+00 seconds.' |  |

Section 5.1

Key Concepts: In this section we examine the idea of a matrix expansion and develop the two-dimensional DFT and inverse DFT as a means of computing the Fourier matrix expansion.

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> A=[1,2;3,4]  >> G00=[1,1;0,0]  >> G01=[1,-1;0,0]  >> G10=[0,0;1,1]  >> G11=[0,0;1,-1]  >> TransformMatrixD(A,G00,G01,G10,G11) | A =  1 2  3 4  G00 =  1 1  0 0  G01 =  1 -1  0 0  G10 =  0 0  1 1  G11 =  0 0  1 -1  D =  1.5000 -0.5000  3.5000 -0.5000 |

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> A=[1,2;3,4]  >> D=(1/4)\*fft2(A) | A =  1 2  3 4  D =  2.5000 -0.5000  -1.0000 0 |

|  |  |
| --- | --- |
| Input Commands | |
| >> A=[250,255,255,252;239,246,254,255;227,237,251,255;149,243,255,253];  >> D=(1/16)\*fft2(A)  D =  2.4225 + 0.0000i -0.0938 + 0.0213i -0.0725 + 0.0000i -0.0938 - 0.0213i  0.0262 - 0.0587i 0.0113 - 0.0700i 0.0075 - 0.0525i 0.0125 - 0.0437i  0.0550 + 0.0000i 0.0575 - 0.0025i 0.0525 + 0.0000i 0.0575 + 0.0025i  0.0262 + 0.0587i 0.0125 + 0.0437i 0.0075 + 0.0525i 0.0113 + 0.0700i | |
| Input Commands | |
| >> A=[250,255,255,252;239,246,254,255;227,237,251,255;149,243,255,253];  >> F=1/4\*[1,1,1,1;1,-i,-1,i;1,-1,1,-1;1,i,-1,-i];  >> D=F\*A\*F'  D =  1.0e+02 \*  2.4225 + 0.0000i -0.0938 - 0.0213i -0.0725 + 0.0000i -0.0938 + 0.0213i  0.0262 - 0.0587i 0.0125 - 0.0437i 0.0075 - 0.0525i 0.0113 - 0.0700i  0.0550 + 0.0000i 0.0575 + 0.0025i 0.0525 + 0.0000i 0.0575 - 0.0025i  0.0262 + 0.0587i 0.0113 + 0.0700i 0.0075 + 0.0525i 0.0125 + 0.0437i | |
| Input Commands | Output (Plot if Applicable) |
| >> A=[1,2;3,4];  >> D1=fft(A);  >> D2=fft(D1');  >> D=1/4\*D2' | D =  2.5000 -0.5000  -1.0000 0 |