PreLab
ELEC6631: Video Processing

Professor: M. Amer, Concordia University, Electrical and Computer Engineering

January 8, 2019

Outline: 2D signals; FT; sampling and reconstruction; effects of aliasing.
The sampling (Nyquist) theorem states that for perfect reconstruction of a band-limited signal, we must sample at a rate (known as the Nyquist rate) equal or above twice the maximum frequency of the original signal. *Over-sampling* is when we exceed the Nyquist rate. *Under-sampling* is when we sample at a lower rate and *critical sampling* is when we sample using exactly the Nyquist rate.

Q1 ——————————————————————————————— (**80 marks**)

Develop a Matlab program that implements the following tasks:

1. Define the 2D *sin* signal, $Z(x, y)$. You can use $f_x = 10\ c/ph$ and $f_y = 10\ c/ph$ for the frequency, $A = 5$ for the amplitude, and $\phi = 0$ for the phase. This is done using

```
A = 5; fx = 10; fy=10; phi=0;
NumberOfSamples = 256;
StepSize = 1/NumberOfSamples;
x=0:StepSize:1-StepSize;% 256 x values from 0 to 1 in steps of 1/256
y=0:StepSize:1-StepSize;% 256 x values from 0 to 1 in steps of 1/256
[X,Y]=meshgrid(x,y);
Z=A*sin(2*pi*fx*X+2*pi*fy*Y+phi);%sinusoidal signal
imshow(Z);
```

2. Use the intensity plot to verify that you have 10 c/ph on the vertical and horizontal axes.
3. Recall that the discrete-time Fourier transform (DTFT) of a sine function is

$$DTFT\{\sin(an)\} = \frac{\pi}{i}\left[\delta(\omega - a) - \delta(\omega + a)\right]. \quad (1)$$

   Use the procedure in the prelab and lab 1 to create intensity and perspective plots of the discrete Fourier transform (DFT) of Z, but utilize Matlab's `imview(DFT)` instead of `imshow` for intensity plotting. Find the pixel location with the highest value. Subtract from the location(s) the center location of the figure (129,129). Verify that you obtain two spikes (i.e., white pixels) at (10,10) and (-10,-10). Now, use your findings to comment on the properties of the spectrum. For example:
   - Is the signal band-limited or not? if so, what is the maximum frequency?
   - Do you see one or two spikes in the spectrum? why is this the case?
   - Has the DFT revealed that we, indeed, have $f_x = 10\ c/ph$ and $fy = 10\ c/ph$?
   - Is our result with the DFT in agreement with what we know about the spectrum of a sine function (e.g., in (1))?
4. Use the maximum frequency and the step-size from part 1 to determine the Nyquist sampling rate. Verify that it is 12.8.
5. Over-sample Z, plot the DFT of the overs-sampled signal, and reconstruct the original signal from the over-sampled signal using *the sinc interpolator* given at the end of the lab. Report your observations. For example, do the DFT and visual inspection still give $f_x = 10\ c/ph$ and $fy = 10\ c/ph$. Why is this the case? Do our findings agree with the Nyquist theorem?

6. Critically-sample Z, plot the DFT of the critically-sampled signal, and do the reconstruction. Note your observations.
7. Under-sample Z, plot the DFT of the under-sampled signal, and reconstruct the original signal from the under-sampled signal using *the sinc interpolator*. What are your observations? For example, do we still have $f_x = 10\ c/ph$ and $fy = 10\ c/ph$ or has the frequency changed or assumed a different "alias". What is this phenomenon?

Solution:

Listing 1: Lab 2 Q1

```
clear;clc;
A = 5; fx = 10; fy=10; phi=0;
NumberOfSamples = 256;
StepSize = 1/NumberOfSamples;
x=0:StepSize:1-StepSize;% 256 x values from 0 to 1 in steps of 1/256
y=0:StepSize:1-StepSize;% 256 x values from 0 to 1 in steps of 1/256
[X,Y]=meshgrid(x,y);
Z=A*sin(2*pi*fx*X+2*pi*fy*Y+phi);%sinusoidal signal

% Finding the DFT
DFT = log(abs(fftshift(fft2(Z))));

% Intensity plot of the DFT
imview(DFT);
% Perspective plot of the DFT
mesh(DFT); colormap hot;

% We note that we have spikes at locations (139,139) and (119,119). by
% subtracting from the center location (129,129) to account for the
% fftshift , we get two spikes at (-10,-10) and (10,10). This agrees
% with what we know about the spectrum of sinusoids. Two spikes at
% locations given by the frequency appearing as images of each other
% around the center point. Our findings here agree with the visual
% inspection and with what we have created ourselves by setting
% fx=10 and fy=10.

% We note also that the signal is band-limited with the maximum
% horizontal and vertical frequencies being 10 and 10. Because
% of this, nearly perfect reconstruction is possible. Neglecting
% roundup and calculation errors.

% The calculation of the Nyquist rate using the maximum frequency
% and the stepsize is as follows
MaximumFrequency = 10;
NyquistFrequency = 2*MaximumFrequency;
NyquistPeriod = 1/NyquistFrequency;

% Now we need to see how many steps we need to take to cover this
```

```
% period.

StepsToCoverPeriod = NyquistPeriod/StepSize

% So practically, we need to take a sample every 12 pixel locations
% or less when we sample in order to obtain the original signal upon
% reconstruction.

% We choose 15 for undersampling, 12 for critical sampling and 4 for
% oversampling.

UnderSampled = Z(1:15:256,1:15:256);
CriticallySampled = Z(1:12:256,1:12:256);
OverSampled = Z(1:4:256,1:4:256);


UnderSampled_DFT = log(abs(fftshift(fft2(UnderSampled))));
CriticallySampled_DFT = log(abs(fftshift(fft2(CriticallySampled))));
OverSampled_DFT = log(abs(fftshift(fft2(OverSampled))));


Recons_From_UnderSampled = sincinterp(UnderSampled,256,256);
Recons_From_CriticallySampled = sincinterp(CriticallySampled,256,256);
Recons_From_OverSampled = sincinterp(OverSampled,256,256);


subplot(4,3,1); imshow(Z);
xlabel('Original');
subplot(4,3,2); imshow(DFT,[]);
xlabel('Original Spectrum');
subplot(4,3,3); mesh(DFT); colormap hot;

subplot(4,3,4); imshow(UnderSampled);
xlabel('Undersampled');
subplot(4,3,5); imshow(UnderSampled_DFT,[]);
xlabel('Undersampled Spectrum');
subplot(4,3,6); imshow(Recons_From_UnderSampled);
xlabel('Recons. from UnderSampled');

subplot(4,3,7); imshow(CriticallySampled);
xlabel('Critically-Sampled');
subplot(4,3,8); imshow(CriticallySampled_DFT,[]);
xlabel('Critically-sampled Spectrum');
subplot(4,3,9); imshow(Recons_From_CriticallySampled);
xlabel('Recons. from Critically sampled');

subplot(4,3,10); imshow(OverSampled);
xlabel('Oversampled');
```

```
subplot(4,3,11); imshow(OverSampled_DFT,[]);
xlabel('Oversampled Spectrum');
subplot(4,3,12); imshow(Recons_From_OverSampled);
xlabel('Recons. from Oversampled');


% We notice that when we under-sample the signal, the frequency has
% certainly changed and is now (7,7) instead of (10,10). The frequencies
% that we see are the aliased component and not the real frequency of the
% signal which we know is 10. This is both evident from the DFT and from
% visually inspecting the reconstructed signal (second row).

% When we critically sample or over-sample we still get the frequency
% as 10 and are able to more or less reconstruct the original signal.
% This agrees with what the sampling theorem tells us.
```

Q2 **(20 marks)**

Develop a Matlab program to:

1. Read a real-world image into Matlab.
2. Create an intensity plot of the DFT of the image.
3. Is the image signal band-limited? Meaning, is there a maximum frequency after which all other frequencies are zero?
4. What does that tell us about sampling/reconstruction of this image signal (i.e., is it possible to perfectly reconstruct the original image from the sampled one)?
5. Prove your finding in the previous step by sampling the signal with a step size of 2 (reduce the size by 1/2) and then reconstruct the signal. Identify areas in the image where aliasing is visible.
6. What is the solution to avoid aliasing?

Note that you can find real-world images for this question under the course website. As students will randomly and independently select an image for this question, it is expected that most students will have used different images.

Solution:

Listing 2: Lab 2 Q2

```
% Reading the image
I = imread('cameraman.tif');
% Generating intensity plot of the DFT
DFT = log(abs(fftshift(fft2(im2double(I)))));

subplot(2,2,1); imshow(I);
xlabel('Cameraman image');

subplot(2,2,2); imshow(DFT,[]);
xlabel('Spectrum');

% Sampling the image
```

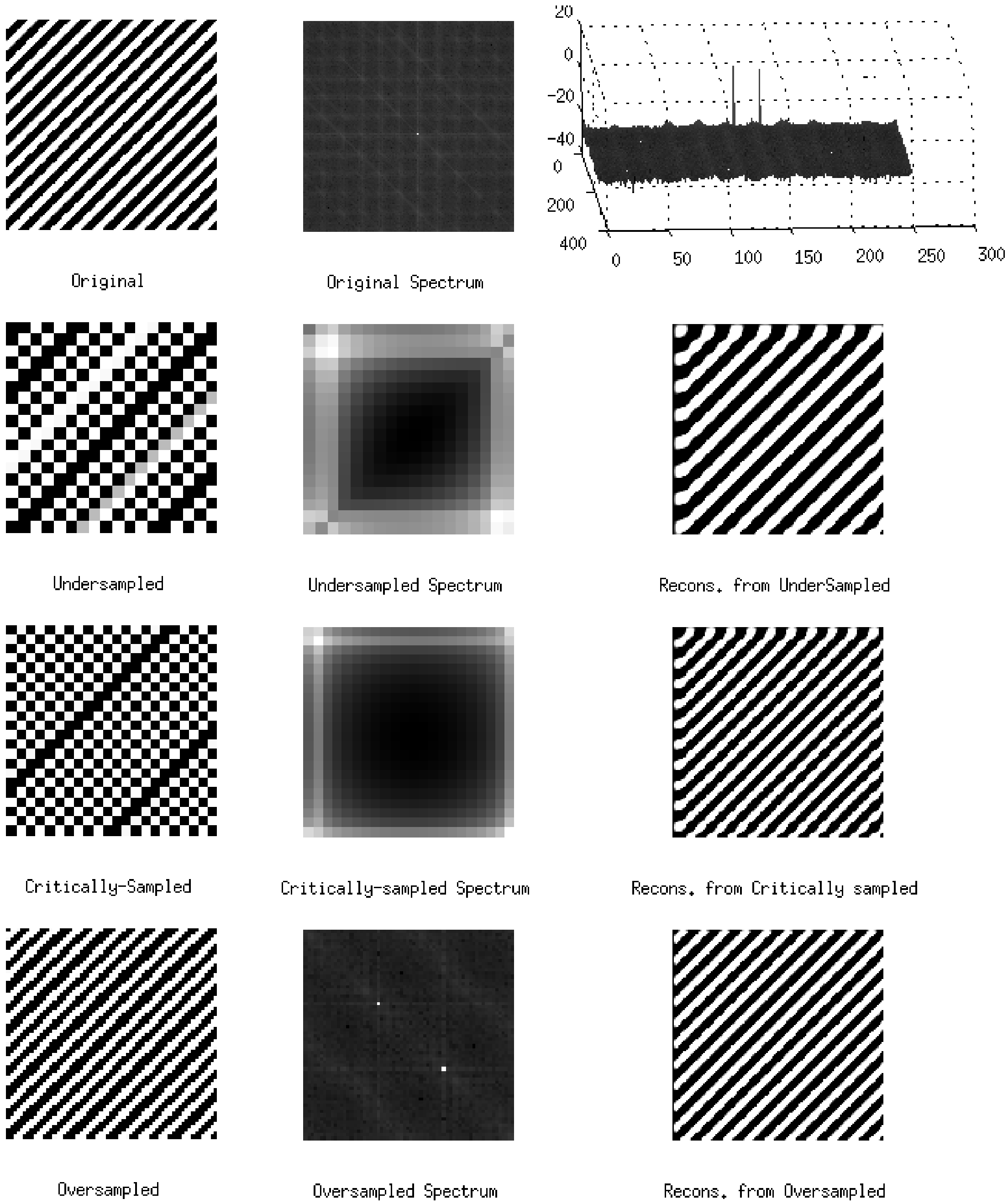

Figure 1: Testing the sampling theorem on a 2D *sin* function.

```
Sampled = I(1:2:256,1:2:256);
subplot(2,2,3); imshow(Sampled,[]);
xlabel('Sampled Cameraman');

% Reconstruction from sampled image using sinc
Recons = sincinterp(im2double(Sampled),256,256);
subplot(2,2,4); imshow(Recons,[]);
```

```
xlabel('Recons. Cameraman');

% From the DFT plot (Fig. 2), as the signal is clearly not band-limited,
% we are bound to have aliasing. This is visible in many areas of
% the reconstructed image. We also notice that the sampled image lost
% a lot of the high frequency information near the edges. This loss
% is bound to happen. However, to avoid aliasing (injecting frequency
% information which is not in the image itself), we can band-limit the
% signal first with a low-pass filter, accept certain degree of loss
% of high frequency information, and reconstruct a better signal
% (aliasing-free).
```

Listing 3: sincinterp.m

```
function Zrecon = sincinterp(Zsampl,Width,Height)
% Zsampl is the sampled signal
% Width is the desired width of the reconstructed signal
% Height is the desired height of the reconstructed signal
% Zrecon is the reconstructed signal
Zrecon = zeros(Height,Width);
[Nx,Ny] = size(Zsampl);
x=linspace(-Nx/2,Nx/2,Width);
y=linspace(-Ny/2,Ny/2,Height);
for n = 1:Nx,
for m = 1:Ny,
          den = pi*(x - (n-0.5*Nx));
          num = sin(den);
          ind = find(den == 0);
          if ~isempty(ind)
              den(ind) = 1; num(ind) = 1;
          end;
          sincx = num ./ den;
          sincx = repmat(sincx, [Height, 1]);
          den = pi*(y - (m-0.5*Ny));
          num = sin(den);
          ind = find(den == 0);
          if ~isempty(ind)
              den(ind) = 1; num(ind) = 1;
          end;
          sincy = num ./ den;
          sincy = repmat(sincy', [1, Width]);
          Zrecon = Zrecon + Zsampl(m, n) .* sincx .* sincy;
     end
end
```

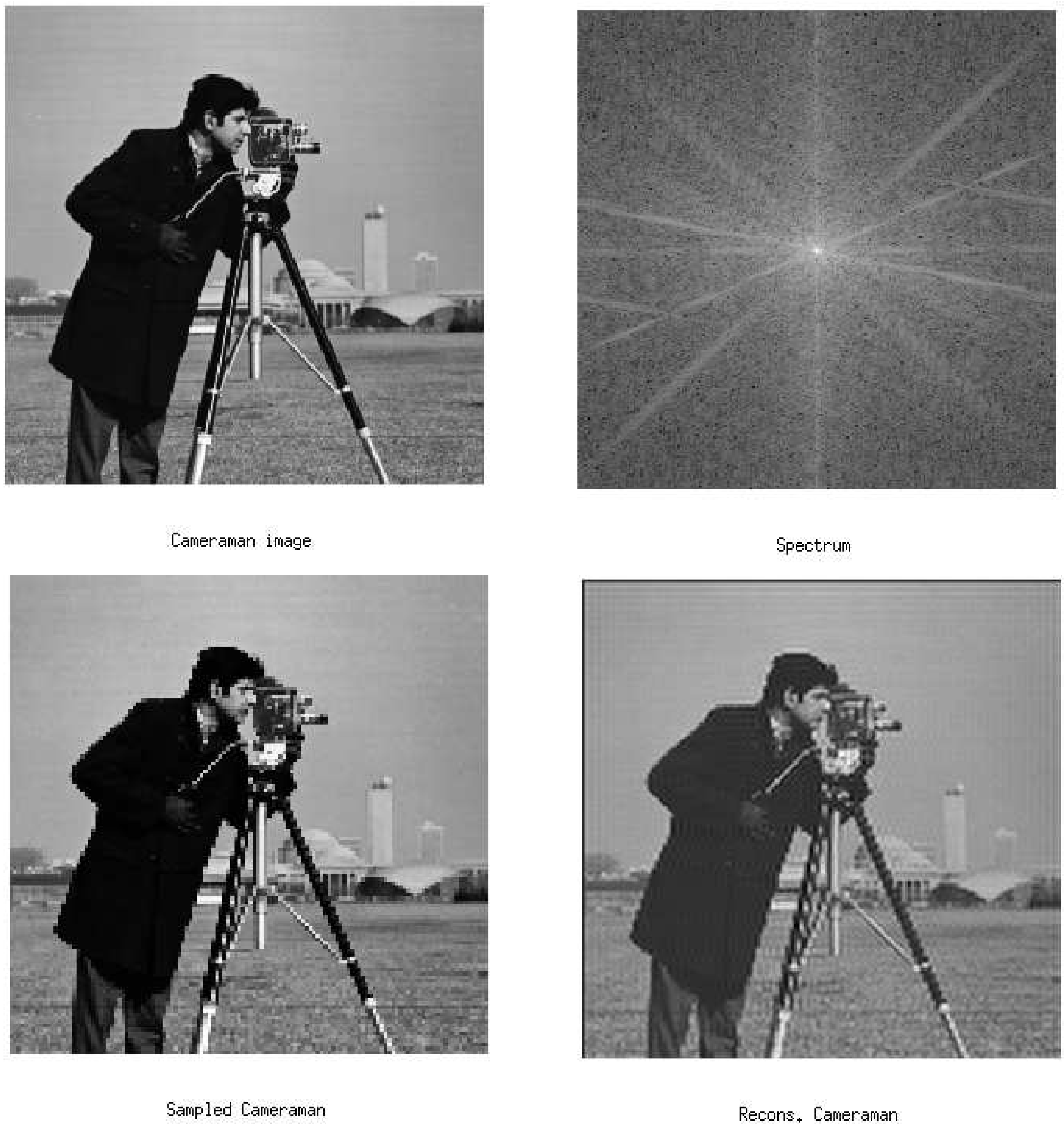

Figure 2: Testing the sampling theorem on a real world image.