

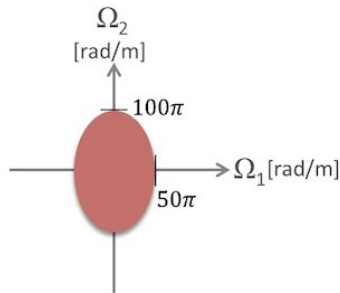
1. Fast Fourier Transform (FFT) speeds up the computation of the Discrete Fourier Transform (DFT) of a signal by compromising the accuracy of the result

1 point

- ☐ True  
☒ False

2. The continuous image  $x(t_1, t_2)$  has the following frequency spectrum. What is the minimum number of samples that should be taken from any  $10\text{cm}$  by  $10\text{cm}$  square region on the image to avoid aliasing?

1 point



- ☐ 5  
☐ 25  
☒ 50  
☐ 100

3. Imagine you want to decrease the resolution of a  $1024 \times 1024$  pixel image by a factor of 2 in each direction so that you end up with a  $512 \times 512$  lower resolution image. Which one of the following procedures should you follow?

1 point

- ☐ low-pass filtering of the original image followed by discarding half of the pixels in the resulting image  
☒ low-pass filtering of the original image followed by discarding three quarters of the pixels in the resulting image  
☐ high-pass filtering of the original image followed by discarding half of the pixels in the resulting image  
☐ high-pass filtering of the original image followed by discarding three quarters of the pixels in the resulting image

4. Imagine now that you want to display a low resolution image on a high resolution TV screen. Which one of the following statements is true?

1 point

- ☐ This can only be done in the spatial domain where the image is taken.  
☐ This can only be done in the frequency domain.  
☒ This can be done in the spatial domain.  
☐ Going from low resolution to high resolution is practically not possible.

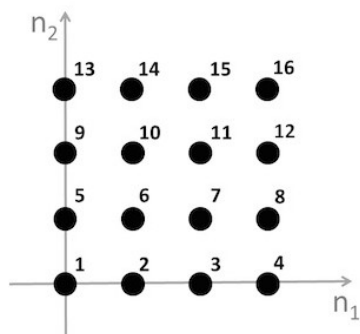
5.  $x(n_1, n_2)$  is defined as  $x(n_1, n_2) = (-1)^{n_1+n_2}$  when  $0 \leq n_1, n_2 \leq 2$  and zero elsewhere. Denote by  $X(k_1, k_2)$ , where  $0 \leq k_1, k_2 \leq 2$ , the DFT of  $x(n_1, n_2)$ . What is the value of  $X(1, 2)$ ?

1 point

- ☐  $1 + j\sqrt{3}$   
☐  $1 - j\sqrt{3}$   
☒ 4  
☐  $(1 + j\sqrt{3})^2$

6. A  $4 \times 4$  pixel image  $x(n_1, n_2)$  is depicted below. Denote by  $X(k_1, k_2)$  the  $4 \times 4$  point DFT of  $x(n_1, n_2)$ . Calculate  $X(0, 0)$

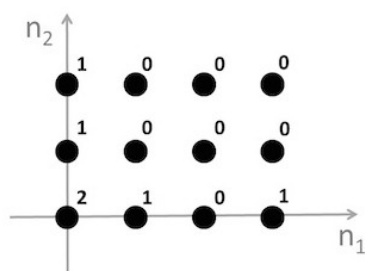
1 point



136

7. Check all statements that are true regarding the  $4 \times 3$  point DFT of the image  $x(n_1, n_2)$  shown below

1 point



- ☒  $X(k_1, k_2)$  is real
- ☐  $X(1, k_2) = X(2, k_2)$  for  $0 \leq k_2 \leq 2$
- ☐  $X(k_1, 0) = X(k_1, 2)$  for  $0 \leq k_1 \leq 3$
- ☐  $X(k_1, k_2) = X(k_2, k_1)$  for  $0 \leq k_1, k_2 \leq 2$
- ☒  $X(0, 0) = 6$



8. In this problem you will get hands-on experience with changing the resolution of an image, i.e., down-sampling and up-sampling. Follow the instructions below to finish this problem.

1 point

(1) Download the original image from [here](#). The original image is an 8-bit gray-scale image of width 479 and height 359 pixels. Convert the original image from type 'uint8' (8-bit integer) to 'double' (real number).

(2) Recall from the lecture that in order to avoid aliasing (e.g., jagged edges) when down-sampling an image, you will need to first perform low-pass filtering of the original image. For this step, create a  $3 \times 3$  low-pass filter with all coefficients equal to  $1/9$ . Perform low-pass filtering with this filter using the MATLAB function "imfilter" with 'replicate' as the third argument. For more information about low-pass filtering using MATLAB, refer to the programming problem in the homework of module 2.

(3) Obtain the down-sampled image by removing every other row and column from the filtered image, that is, removing the 2<sup>nd</sup>, 4<sup>th</sup>, all the way to the 358<sup>th</sup> row, and then removing the 2<sup>nd</sup>, 4<sup>th</sup>, all the way to the 478<sup>th</sup> column. The resulting image should be of width 240 and height 180 pixels. This completes the procedure for image down-sampling. In the next steps, you will up-sample this low-resolution image to the original resolution via spatial domain processing.

(4) Create an all-zero MATLAB array of width 479 and height 359. For every odd-valued  $i \in [1, 359]$  and odd-valued  $j \in [1, 479]$ , set the value of the newly created array at  $(i, j)$  equal to the value of the low-resolution image at  $(\frac{i+1}{2}, \frac{j+1}{2})$ . After this step you have inserted zeros into the low-resolution image.

(5) Convolve the result obtained from step (4) with a filter with coefficients  $[0.25, 0.5, 0.25; 0.5, 1, 0.5; 0.25, 0.5, 0.25]$  using the MATLAB function "imfilter". In this step you should only provide "imfilter" with two arguments instead of three, that was the case in step (1). The two arguments are the result from step (4) and the filter specified in this step. This step essentially performs bilinear interpolation to obtain the up-sampled image.

(6) Compute the PSNR between the upsampled image obtained from step (5) and the original image. For more information about PSNR, refer to the programming problem in the homework of module 2. Enter the PSNR you have obtained to two decimal points in the box below.

28.17