Computer Vision
2011/12

Universidade da Beira Interior
Departamento de Informática
2011/12

1. Implement a MATLAB function that receives a set of “d-dimensional” points represented in homogeneous coordinates (one per row) and returns their representation in the Cartesian space.

   Prototype: cartesian homogenous2Cartesian(points)

2. Consider the following problem: Mr. X has two grayscale images and know that image 1 corresponds to the second image rotated by an angle that is multiple of Π/2. Help Mr. X and implement a MATLAB function that returns the correct angle of rotation from image 1 to image 2.

   Prototype: angle findRotation(img1, img2)

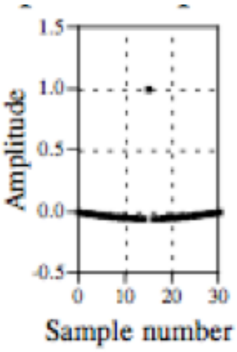
3. Give examples of two 2D functions, when the first one is homogenous (but not aditive, thus not linear), and the second one is linear.

4. Consider a the following 1D signal and obtain the result of the convolution with the following kernel:

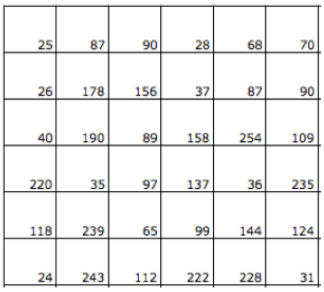
   Signal=[1 -1 0 1 1 -1 1 0 0 1]

   Kernel=[0.5 0 -1]

5. Analyse the following impulse response of a 1D filter. Does it correspond to an high-pass or low-pass kernel? Why?

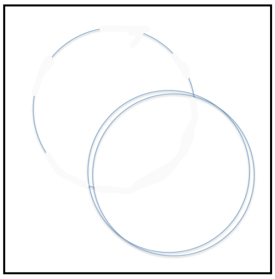


6. Consider the following grayscale image. Obtain the histogram of 10 bins for this image.



25	87	90	28	68	70
26	178	156	37	87	90
40	190	89	158	254	109
220	35	97	137	36	235
118	239	65	99	144	124
24	243	112	222	228	31

7. Consider the following image. Suppose that you implemented a circular Hough transform to find circles of a specified radius in that image. Illustrate an example of the matrix accumulator in the Hough space (with (x,y) as free parameter, as radius is a constant value) that would result of this image.



8. Implement a MATLAB function that receives an image that is the output of the Sobel edge detector and performs non-maximal suppresion, i.e., all pixels that are not maximums in a given radius should be set to 0.

   Prototype: retImg nonMaxSup(img, radius)

9. Implement a MATLAB function to verify if two chain codes are equivalent, meaning that they were extracted from the same contour. Obviously, these codes are not normalized (0=no, 1=yes). We consider 4-neighborhood here.

   Prototype: yn chainCorresponding(code1, code2)

10. Implement a MATLAB function that receives an image, a set of vectors (one per row) and returns only those that are eigenvectors of the image.

    Prototype: ret eigenVectors (img, vectors).

11. Suppose that the following matrix is the output from the MATLAB "princomp" function, meaning that it transforms data in an original space to the condensed PCA version. Based on the following set of original data (1 instance per row), write the PCA consensed version of this data.

PCA MATRIX

1	**0**	**1**
-1	**1**	**1**
1	**-1**	**-1**
1	**0**	**0**

Original Data: [10 20 -10 0; 10 -20 -10 10; 0 0 -10 10; 20 10 10 -10];