ICBV231 HW 2

Release Date: 15/12/2022

Submission Deadline: 28/12/2022, 23:59

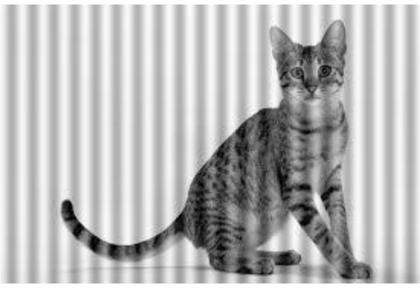
Instructions

Please read and follow these instructions carefully.

- To be able to submit your work you must first enroll in a group on Moodle.
- The assignment contains both programming tasks and a written part. The written part needs to be submitted in a single PDF file and the programming tasks must be answered in the provided Jupyter notebook. You need to submit a single zip file where:
    - The name of the file is "Assignment2-Group#.zip", where # should be replaced by your group number.
    - In the zip file put the written answers as a PDF file and the Jupyter notebook (.ipynb file) along with any additional files used by your code.
- The written assignment must be typed on the computer. Handwritten scanned documents **will not** be accepted.
- Make sure that the Jupyter notebook can be run. Before submitting, restart runtime and run all cells. In Google Colab: 1) *Runtime > Restart and run all;* 2) *File > Download .ipynb*. In addition, make sure you submit any additional files used by your code (usually image files).
- Both the PDF containing the written answers and the Jupyter notebook must contain the id numbers and the names of the submitting students.

Question 1 – Convolution, Fourier transform and linear systems

Section A – DFT "Free Willy"



Willy the cat found himself in a cage, behind bars. Your mission is to release him. In order to do so, you should read the attached image **cat_bars.png** and use what you learned about Fourier transform and filtering of spatial frequencies to create an image where Willy is free (i.e., removing the bars while leaving the image as intact as possible). You can assume the bars are a wave of a single frequency.

Guidance: Compute the image DFT using the numpy.fft.fft2 function. Shift the zero-frequency component to the center of the spectrum using numpy.fft.fftshift. Use the amplitude spectrum to locate and equate to zero the Fourier coefficients corresponding to the relevant frequency of the vertical bars. Free Willy by applying the inverse DFT on the filtered Fourier spectrum using numpy.fft.ifft2.

Plot the following computational steps (i.e., a total of 5 plots):

1. The original grayscale image.
2. The image amplitude spectrum.
3. The binary filter (mask) in which you multiplied the Fourier spectrum in order to equate to zero the Fourier coefficients you chose.
4. The amplitude spectrum after the filtering.
5. The filtered image (i.e., the inverse DFT of the filtered Fourier spectrum).

Is your reconstruction of the details "behind the bars" perfect? why?

Section B – 2D discrete convolution

You are given the following 2 kernels:

$G =$

0.5	1	0.5
1	6	1
0.5	1	0.5

$F =$

-0.5	-1	-0.5
-1	6	-1
-0.5	-1	-0.5

And the following 2D 7x7 image:

$I =$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

1. What can the kernels G and F be used for in the context of vision (computational and/or biological)?
2. What is the name of the type of signal image I is?
3. Compute $I * G$. Your response should be a 7x7 image.
4. Compute $F * (I * G)$. Your response should be a 7x7 image.
5. Design a kernel H (5x5) such that your previous response will be equal to $I * H$.

Question 2 – Edge detection

Section A

You are given the following grayscale image:

$$I(x, y) = x + \sin(y)$$

Let E_1 be an edge detector based on finding the local maximum of $|\nabla I|$ and E_2 be an edge detector based on finding the zero-crossing of the Laplacian ΔI. Write the mathematical expressions of the edge points detected by each of the operators. Are the results identical?

Guidance: Compute analytically the edge points detected by each of the edge detectors when applied to the image. For E_1 this means you should first find the expression of $|\nabla I(x, y)|$ (the gradient magnitude), and then find the local maximum points of $|\nabla I(x, y)|$. Recall that in order to find extremum points you should compute the first derivative of the expression you found for $|\nabla I(x, y)|$ and equate it to zero. Then, you compute the second derivative of the expression you found for $|\nabla I(x, y)|$, and compute its value at the extremum points you previously found. If the value of the second derivative at this point is greater than 0, then it is a local minimum. If the value of the second derivative at this point is smaller than 0, then it is a local maximum. For E_2 this means you should first find the expression of the Laplacian $\Delta I(x, y)$. Then, you should find all the points where $\Delta I(x, y) = 0$.

Section B

In practical session #6 you have been shown an implementation of the LoG (this implementation is also provided in the attached Jupyter notebook). Locating edges via finding values close to zero in the Laplacian is too noisy. Hence, you will implement an edge detector that uses a stricter criterion of zero-crossing. Let LoG denote the Laplacian of Gaussian for an image I. For some threshold values $a \geq 0 \geq b$, a pixel (p_x, p_y) is considered a zero-crossing pixel if one of the following is true:

- $LoG[p_x, p_y] > a$ and a pixel (p'_x, p'_y) exists in the 8-environment of (p_x, p_y) so that $b > LoG[p'_x, p'_y]$.
- $b > LoG[p_x, p_y]$ and a pixel (p'_x, p'_y) exists in the 8-environment of (p_x, p_y) so that $LoG[p'_x, p'_y] > a$.

Your goal is to detect the edges of the coins present in the attached **coins.jpg** image. Experiment with the values of *a, b* and the parameters of the Gaussian used for smoothing, to find the values that generate the best result on the provided coins image (evaluate the quality of the result by its appearance). Plot the following computational steps (i.e., a total of 4 plots):

1. The original image in grayscale.
2. The smoothed image (i.e., the image after applying the Gaussian smoothing).
3. The image after applying LoG.
4. The image of the detected edges after applying zero-crossing.

Note: The 8-environment of a pixel (p_x, p_y) is defined as: $(p_x - 1, p_y)$, $(p_x - 1, p_y - 1)$, $(p_x, p_y - 1)$, $(p_x + 1, p_y - 1)$, $(p_x + 1, p_y)$, $(p_x + 1, p_y + 1)$, $(p_x, p_y + 1)$, $(p_x - 1, p_y + 1)$.

Section C

In this section you are provided with the same coins image with Gaussian noise added to it (in the attached Jupyter notebook the noisy image is called '*noisy_coins*'). Find the edges of the coins in this image using the zero-crossing method you created in section B, and using OpenCV implementation of Canny's algorithm. Try to find good parameters for both methods. Display, side by side, the best result you achieved on the noisy image using your method and Canny's algorithm. Which method produced better results? Why do you think this is the case?

Question 3 – Parametric curves

Section A

In class you saw how a parametric curve can be described as mapping from an interval to the image plane:

$$\alpha(t) = \big(x(t), y(t)\big)$$

A 3D curve can be described in the same way, as mapping from an interval to a 3D space:

$$\alpha(t) = \big(x(t), y(t), z(t)\big)$$

Use this description to prove that a perspective image of a straight line in the 3D world is a straight line in the image plane.

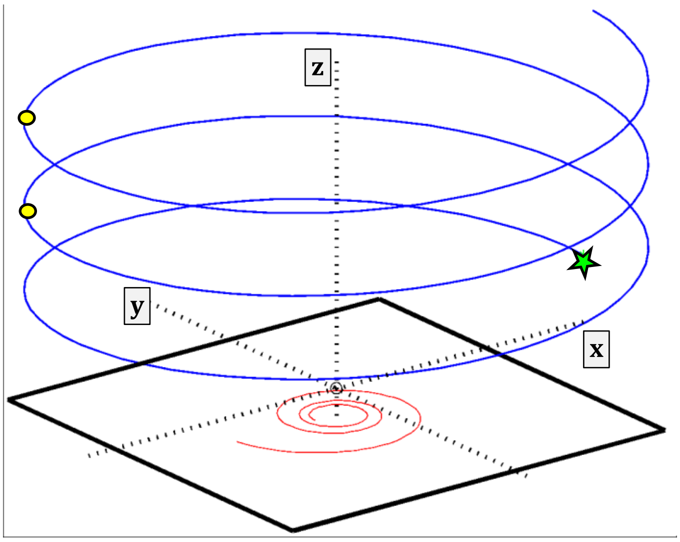
Reminder:

A parametric description of a line in the plane is $\alpha(t) = (x_0, y_0) + t \cdot (a_1, a_2)$, and a parametric description of a line in space is $\alpha(t) = (x_0, y_0, z_0) + t \cdot (a_1, a_2, a_3)$.

Section B

The image plane of a pinhole camera is parallel to the XY plane, its pinhole is located at the world's origin, and its focal length is $f = 2$. The following graph presents an object in the world (painted in blue), which is the trace of a Helix curve (a "corkscrew" curve) $\alpha(s)$. This curve starts at the world point (1,0,5) (marked by a green star). In addition, the graph presents the trace of the curve $\beta(s)$ (painted in red), which is the perspective projection of $\alpha(s)$ on the image plane.

Assume that $\alpha(s)$ is centered around the Z axis, and that the vertical distance between the two yellow dots is 1. Examine the object and its projection, and answer the following questions:



1. Suggest a parametrization for $\alpha(s)$, and define the interval s ranges over.
2. Is $\alpha(s)$ regular?
3. Compute a parametric description for $\beta(s)$, based on $\alpha(s)$ you suggested.

Question 4 – Reading material

Answer the following question in writing and <u>keep your answer short</u>. This question refers to *What the frog's eye tells the frog's brain*, by Lettvin, Maturana, McMculloch, and Pitts, 1960.

In class, we discussed whether the human visual system, or more specifically the retinal response, is a linear system. As we cannot use invasive methods to study the human eye, maybe we could answer this question by studying the visual system of other less complex animals.

Locate in the mentioned article <u>at least</u> two findings which are indicative of nonlinear actions (nonlinear spatial response) in the frog's eye. For each of these findings specify the corresponding paragraph in the article which discusses the finding and write which part in the graph supports the claim that the finding is indicative of a nonlinear action. Explain why the action is nonlinear.