Melika Mohammadi Fakhar - 99522086

1.
1.1
Gradient in the x-direction (horizontal direction):

$$\frac{\partial I}{\partial x} = I_x$$

Gradient in the y-direction (vertical direction):

$$\frac{\partial I}{\partial y} = I_y$$

The gradient vector at any point (x,y) on the image can be represented by combining the gradients in both directions into a vector form: $\vec{G} = (I_x, I_y)$
This vector points in the direction of the steepest ascent of the image intensity surface at the point (x,y).

1.2
Calculating the gradient vector of an image is crucial for various image processing and computer vision tasks because it offers detailed insights into changes in image intensity. This information is instrumental in:

- **Edge Detection:** Identifying boundaries between different regions based on changes in intensity.
- **Feature Extraction:** Analyzing patterns, textures, and orientations for recognition tasks.
- **Image Enhancement:** Sharpening images by emphasizing edges.
- **Motion Detection and Analysis:** Understanding motion in video through changes over time.

- **Noise Reduction:** Smoothing out noise while preserving important edge information.

1.3

$$Magnitude = |\vec{G}| = \sqrt{I_x{}^2 + I_y{}^2}$$

1.4

$$\theta = \arctan\left(\frac{I_y}{I_x}\right)$$

1.5

The process unfolds through several key steps:

**Noise Reduction:** Utilizes a Gaussian filter to smooth the image and mitigate noise.
**Gradient Calculation:** Employs Sobel operators (or similar) to compute the gradient magnitude and direction for each pixel, highlighting potential edges.
**Non-maximum Suppression:** Thins out edges by retaining only the strongest edge pixels, based on the gradient direction.
**Double Thresholding:** Differentiates between strong, weak, and non-edges using two thresholds, to refine edge detection.
**Edge Tracking by Hysteresis:** Ensures continuity and connection of edges by preserving weak edges that are connected to strong edges.

Advantages of the Canny detector include its **robustness against noise, precise edge localization, ability to provide a single response per actual edge, and adaptability to different imaging scenarios**. These strengths make it superior in accuracy and reliability to many other edge detection methods,

positioning it as a preferred choice for a wide range of applications.

1.6

**Sensitivity to Noise:** The Laplacian operator's second-order derivative nature makes it more sensitive to noise, potentially resulting in false edge detections in real-world images. On the other hand, Sobel and Canny operators, utilizing first-order derivatives, are less affected by noise. The Canny method further reduces noise impact through pre-edge detection Gaussian filtering.

**Non-directional**: The Laplacian operator's isotropic approach detects edges in all directions without indicating their orientation, which can be a limitation for detailed analysis. In contrast, gradient-based methods like the Sobel operator discern edge orientation by calculating derivatives in specific directions, offering richer edge information in structured images.

**Single Response to Edges**: The Laplacian operator's single response per edge results in thinner detections and struggles with wide or blurred edges due to its inability to handle gradual intensity changes. This makes edges less prominent and difficult to track in complex images. Conversely, the Canny detector effectively identifies both strong and weak edges, offering more robust and traceable edge detection.

4.

A.

- **Image Filtering and Convolution:**

*Method/Tool:* Fourier Transform-based Convolution.
*Explanation:* In image processing and computer vision, convolution operations are essential for tasks such as edge detection, blurring, and feature extraction. However, performing

convolution directly in the spatial domain can be computationally expensive, especially for large filters and high-resolution images. Fourier Transform offers a solution by enabling convolution in the frequency domain. By convolving the Fourier transforms of the image and the filter, and subsequently taking the inverse Fourier transform of the result, the convolution operation can be efficiently executed, significantly speeding up the process.

- **Image Compression:**

*Method/Tool:* JPEG Compression.
*Explanation:* JPEG compression, a widely used image compression technique, utilizes Fourier analysis extensively. It employs the Discrete Cosine Transform (DCT), a variant of the Fourier Transform, to convert image data from the spatial domain to the frequency domain. In the frequency domain, the image is represented by its frequency components. Since natural images tend to have most energy concentrated in low-frequency components, the DCT enables efficient compression by discarding higher-frequency components with less visual significance. By quantizing and compressing these frequency coefficients, JPEG achieves high compression ratios while preserving acceptable image quality.

- **Image Registration and Alignment:**

*Method/Tool:* Cross-Correlation using Fourier Transform.
*Explanation:* Image registration involves aligning multiple images of the same scene captured from different viewpoints, times, or sensors. A common technique for image registration is cross-correlation, where the similarity between images is measured by sliding one image over another and computing their similarity at each position. This process can be computationally intensive, especially for large images. However, by utilizing Fourier Transform to compute cross-correlation in the frequency domain, computational complexity can be reduced. The Fourier Transform of the cross-correlation is efficiently computed, and the peak of the result indicates the optimal alignment between the images.

This method finds extensive use in medical imaging, remote sensing, and panoramic image stitching applications.

B.

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

$F(0,0)$ for an image represents the DC (Direct Current) component or the average intensity of the image. When we calculate the Fourier transform of an image and evaluate it at the origin (u=0,v=0), we're essentially looking at the frequency component at the lowest possible frequencies in both the horizontal and vertical directions.

6.
We can calculate the number of iterations needed using the formula derived from the RANSAC's probabilistic model:

$$N = \frac{\log(1-p)}{\log(1-w^n)}$$

where:
- N is the number of required iterations,
- p is the desired probability of success (0.99),
- w is the proportion of inliers in the data (0.4),
- n is the minimum number of points needed to estimate the model parameters (for a circle, n=3).

**Approximately 70 repetitions are needed.**

7.
a.
**Concept and Methodology:** The Hough Transform uses a voting procedure in a parameter space to detect shapes, making it good for identifying lines in noisy images or when lines are not perfectly straight. LSD, on the other hand, operates directly on the image pixels, detecting gradient changes to find line segments, which makes it precise in identifying the start and end points of lines.

**Computational Efficiency:** The Hough Transform can be computationally demanding, especially for high-resolution images or when seeking high precision, due to its reliance on an accumulator space. LSD is generally more efficient because it does not require a parameter space or accumulator array, aligning it better with real-time applications.

**Accuracy and Robustness:** While the Hough Transform is robust against image noise and can detect incomplete or slightly curved lines, its accuracy is contingent on the granularity of the parameter space. LSD excels in accurately detecting line segments, including precise endpoints, though it might be more sensitive to image noise and might not detect very short or obscured lines as effectively.