

# Image Processing Techniques

## Sobel Filter

The Sobel filter is a basic tool for finding edges in an image. It works by looking at how the brightness of the image changes in two directions: horizontally and vertically. The Sobel filter uses two small grids, called kernels, to detect these changes. After applying these grids, it combines the results to highlight the edges. This filter is often used when you want to detect outlines or borders in pictures.

```
import cv2

import numpy as np

image = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)

sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)

sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)

sobel_magnitude = np.sqrt(sobel_x**2 + sobel_y**2)

cv2.imshow('Sobel Magnitude', sobel_magnitude)

cv2.waitKey(0)

cv2.destroyAllWindows()
```

## Laplacian Filter

The Laplacian filter is another method for finding edges in images, but it works differently from the Sobel filter. Instead of looking for changes in specific directions, it detects where the image changes rapidly in any direction. This filter is useful for finding all edges in an image, regardless of direction.

```
import cv2
```

```
image = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
```

```
laplacian = cv2.Laplacian(image, cv2.CV_64F)
```

```
cv2.imshow('Laplacian', laplacian)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

## **Canny Edge Detector**

The Canny Edge Detector is a popular tool for finding clear and accurate edges in images. It first smooths the image to reduce noise, then looks for edges by checking where the image changes quickly. It also applies some extra steps to make sure only the most important edges are kept. The Canny Edge Detector is widely used because it gives reliable results.

```
import cv2
```

```
image = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
```

```
edges = cv2.Canny(image, threshold1=100, threshold2=200)
```

```
cv2.imshow('Canny Edges', edges)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

## **Contours in Image Processing**

Contours are lines that follow the shapes in an image. They are useful for identifying and isolating objects. By finding contours, you can detect and outline different shapes, which is helpful for tasks like object recognition. To find contours, you usually convert the image to black and white first, then trace the outlines of objects.

```
import cv2
```

```
image = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
```

```
_, binary = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
```

```
contours, _ = cv2.findContours(binary, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

```
contoured_image = cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)
```

```
cv2.drawContours(contoured_image, contours, -1, (0, 255, 0), 2)
```

```
cv2.imshow('Contours', contoured_image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```