

Report on Filters and Edge Detection

1 Sobel Filter

The Sobel filter is a gradient-based edge detection method used to highlight edges in images. It calculates the gradient of the image intensity at each pixel, emphasizing regions of high spatial frequency that correspond to edges. The Sobel operator uses two 3x3 convolution kernels: one for detecting changes in the horizontal direction (Sobel X) and one for the vertical direction (Sobel Y). The resulting gradient magnitude is calculated by combining these two gradients.

Python Code:

```
import cv2
import numpy as np

# Load an image in grayscale
image = cv2.imread('image.png', cv2.IMREAD_GRAYSCALE)

# Apply Sobel filter
sobel_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3) # Horizontal edges
sobel_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3) # Vertical edges

# Calculate the gradient magnitude
sobel_combined = cv2.magnitude(sobel_x, sobel_y)
```

The Sobel filter is used in edge detection, object detection, and feature extraction, where identifying the edges of objects is crucial.

2 Laplacian Filter

The Laplacian filter is a second-order derivative filter used for edge detection. It measures the rate of change of the gradient by approximating the second derivative of the image intensity, thus highlighting regions with rapid intensity changes. This makes it effective for edge detection but also sensitive to noise.

Python Code:

```
# Apply Laplacian filter
laplacian = cv2.Laplacian(image, cv2.CV_64F)
```

The Laplacian filter is used in edge detection, image sharpening, and feature extraction tasks.

3 Canny Edge Detector

The Canny edge detector is a multi-stage algorithm designed to detect a wide range of edges in images. It involves applying a Gaussian filter for smoothing, finding intensity gradients, performing non-maximum suppression to thin out edges, and using double thresholding to detect and connect edges. The result is a binary image with edges marked in white.

Python Code:

```
# Apply Canny edge detector
canny_edges = cv2.Canny(image, 100, 200)
```

The Canny edge detector is effective and robust, commonly used in object recognition, medical imaging, and computer vision.

4 Contours in Image Processing

Contours are curves joining continuous points along a boundary with the same color or intensity. In image processing, contour detection follows edge detection and is performed using algorithms like Canny or thresholding, followed by the `findContours` function. Contours are useful for shape analysis, object detection, and recognition.

Python Code:

```
# Find contours
contours, hierarchy = cv2.findContours(canny_edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Draw contours on the original image
cv2.drawContours(image, contours, -1, (0, 255, 0), 2)
```

Contours help in extracting features from objects in an image, such as their size, shape, and orientation, and are widely used in object tracking and shape-based image analysis.