

121450036-rika-ajeng-finatih-3

September 1, 2024

Nama : Rika Ajeng Finatih

NIM : 121450036

Kelas : Pengenalan Pola RA

0.1 Implementasi Algoritma Deteksi Tepi Sobel dan Canny dalam Mengidentifikasi Tepi Gambar

Algoritma deteksi tepi Sobel dan Canny adalah dua metode populer dalam pemrosesan citra digital yang digunakan untuk mendeteksi tepi dalam gambar. Deteksi tepi adalah teknik dalam pengolahan citra digital yang bertujuan untuk mengidentifikasi titik-titik pada gambar di mana terjadi perubahan intensitas cahaya secara signifikan. Perubahan ini menandakan adanya batas atau tepi suatu objek dalam gambar.

Implementasi dengan python:

```
[31]: import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow # Import cv2_imshow untuk Google Colab

# Baca citra dalam mode grayscale
img = cv2.imread('/content/logo_sains_data.jpg', 0)

# -----
# Implementasi Algoritma Deteksi Tepi Sobel
# -----
# 1. Deteksi tepi menggunakan operator Sobel
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=3) # Gradien di arah x
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=3) # Gradien di arah y

# 2. Gabungkan hasil Sobel x dan y untuk mendapatkan magnitude gradien
sobel_edges = cv2.magnitude(sobelx, sobely)

# 3. Normalisasi hasil deteksi tepi untuk memastikan nilai berada dalam rentang 0-255
sobel_edges = cv2.normalize(sobel_edges, None, 0, 255, cv2.NORM_MINMAX)
```

```

# 4. Konversi hasil ke format yang sesuai untuk ditampilkan (uint8)
sobel_edges = sobel_edges.astype(np.uint8)

# -----
# Implementasi Algoritma Deteksi Tepi Canny
# -----
# 1. Penghalusan gambar menggunakan Gaussian Blur
blurred_img = cv2.GaussianBlur(img, (5, 5), 1.5)

# 2. Deteksi tepi menggunakan algoritma Canny
canny_edges = cv2.Canny(blurred_img, 50, 150) # Ambang batas bawah dan atas

# -----
# Visualisasi Hasil
# -----
plt.figure(figsize=(12, 6))

# Gambar asli
plt.subplot(1, 3, 1)
plt.title('Gambar Asli')
plt.imshow(img, cmap='gray')
plt.axis('off')

# Deteksi tepi dengan Sobel
plt.subplot(1, 3, 2)
plt.title('Deteksi Tepi Sobel')
plt.imshow(sobel_edges, cmap='gray')
plt.axis('off')

# Deteksi tepi dengan Canny
plt.subplot(1, 3, 3)
plt.title('Deteksi Tepi Canny')
plt.imshow(canny_edges, cmap='gray')
plt.axis('off')

plt.tight_layout()
plt.show()

```



0.2 Interpretasi Hasil Analisis:

Berdasarkan hasil gambar yang diperoleh, dapat diamati beberapa hal sebagai berikut.

1. Deteks Tepi Sobel:

- Tepi-tepi huruf dan gambar gantungan terlihat cukup jelas.
- Ada beberapa noise atau tepi tambahan, terutama di area dengan gradasi warna atau tekstur yang kompleks.
- Terdapat beberapa tepi tipis mungkin kurang terdeteksi dengan baik.

2. Deteksi Tepi Canny:

- Tepi-tepi terlihat lebih halus, tipis, dan akurat dibandingkan Sobel.
- Noise jauh lebih sedikit, menghasilkan tampilan tepi yang lebih bersih.
- Tepi-tepi tipis berhasil dideteksi dengan baik.

Secara keseluruhan, algoritma Canny memberikan hasil yang lebih baik dalam mendeteksi tepi pada logo “SAINS DATA” ini. Tepi-tepi yang dihasilkan lebih akurat, bersih, dan representatif dari batas-batas objek dalam gambar. Hal ini menunjukkan bahwa Canny lebih efektif dalam mengurangi noise dan mendeteksi tepi dengan berbagai ketebalan.