Sultan Fa'iq Athallah
1103213020

Robotika Week 11

1. Analisis Simulasi *Information Extraction* Menggunakan Python dan OpenCV

1.1 Ekstraksi Garis dengan *Hough Transform*

Penjelasan Kode:

- Gambar diubah menjadi *grayscale* untuk mempermudah deteksi fitur.
- Tepi gambar diidentifikasi menggunakan algoritma *Canny*, dengan parameter batas bawah dan atas tertentu.
- Transformasi Hough diterapkan untuk mendeteksi garis berdasarkan informasi tepi.
- Garis yang terdeteksi digambarkan di gambar asli menggunakan koordinat hasil Transformasi Hough.

Analisis Hasil:

- Teknik ini efektif untuk mendeteksi garis lurus yang mencolok.
- Keberhasilannya dipengaruhi oleh kualitas gambar dan pengaturan ambang pada algoritma *Canny*.

1.2 *Template Matching* untuk Deteksi Objek

Penjelasan Kode:

- Gambar *template* dan target dimuat dalam mode *grayscale*.
- Fungsi cv2.matchTemplate digunakan untuk mencocokkan pola *template* dengan gambar target.
- Lokasi kecocokan tertinggi ditandai dengan kotak.

Analisis Hasil:

- Teknik ini sangat berguna untuk mendeteksi pola tertentu.
- Keterbatasannya adalah sensitivitas terhadap rotasi dan perubahan skala objek.

1.3 Pembuatan *Pyramid* Gambar

Penjelasan Kode:

- Gambar diperkecil secara bertahap menggunakan fungsi cv2.pyrDown untuk membentuk piramida multi-resolusi.
- Tiap tingkat menunjukkan gambar dengan resolusi lebih rendah.

Analisis Hasil:

- Berguna untuk analisis skala multi-resolusi, misalnya dalam deteksi fitur pada berbagai tingkat detail.
- Namun, detail penting bisa hilang pada resolusi yang lebih rendah.

1.4 Deteksi Lingkaran dengan *Hough Transform*

Penjelasan Kode:

- Gambar di-*blur* menggunakan *GaussianBlur* untuk mengurangi *noise*.
- Transformasi Hough diterapkan untuk mendeteksi lingkaran berdasarkan gradien tepi.
- Lingkaran yang terdeteksi digambarkan pada gambar asli.

Analisis Hasil:

- Cocok untuk mendeteksi lingkaran yang jelas tanpa banyak *noise*.
- Akurasi menurun jika *noise* tinggi atau bentuk lingkaran terdistorsi.

1.5 Ekstraksi Warna Dominan

Penjelasan Kode:

- Gambar dikonversi ke format RGB dan diubah menjadi array 2D untuk proses *clustering*.
- Algoritma *KMeans* digunakan untuk mengelompokkan piksel menjadi beberapa kluster warna.
- Warna dominan diidentifikasi berdasarkan pusat kluster (*centroid*).

Analisis Hasil:

- Efektif untuk menganalisis warna utama pada gambar.
- Akurasi dipengaruhi oleh jumlah kluster dan distribusi warna dalam gambar.

1.6 Deteksi Kontur Gambar

Penjelasan Kode:

- Tepi objek diidentifikasi menggunakan algoritma *Canny*.
- Fungsi cv2.findContours mendeteksi kontur dari tepi yang ditemukan.
- Kontur digambar pada gambar asli.

Analisis Hasil:

- Sangat berguna untuk mendeteksi bentuk dan batas objek.
- Akurasi bergantung pada deteksi tepi dan parameter algoritma *Canny*.

---

2. Analisis Simulasi Webots

2.1 Ekstraksi Data *Lidar*

Penjelasan Kode:

- Sensor *Lidar* diaktifkan dan dikonfigurasi untuk membaca data rentang jarak dalam bentuk array.
- Fungsi lidar.getRangeImage digunakan untuk mengumpulkan data jarak dari lingkungan.
- Data dianalisis untuk menentukan keberadaan objek di berbagai arah.

Analisis Hasil:

- Data *Lidar* memberikan informasi jarak yang akurat untuk navigasi.
- Berguna dalam navigasi robot dan penghindaran rintangan.

2.2 Deteksi Rintangan

Penjelasan Kode:

- Sensor jarak ultrasonik digunakan untuk mendeteksi keberadaan rintangan di depan robot.
- Data sensor diproses untuk menghitung kecepatan motor kiri dan kanan dengan mempertimbangkan koefisien penghindaran tabrakan.
- Fungsi compute_speeds mengatur kecepatan motor untuk menghindari rintangan.

Analisis Hasil:

- Teknik ini efektif untuk mendeteksi dan menghindari rintangan secara *real-time*.
- Keberhasilan tergantung pada penempatan sensor dan pengaturan parameter koefisien.