

Nama : Jhose Immanuel Sembiring

NIM : 1103202047

Tugas Week 11

1. Ekstraksi Garis Menggunakan Hough Transform

Pada simulasi ini, dilakukan deteksi garis pada gambar menggunakan metode Hough Transform setelah melalui proses deteksi tepi menggunakan Canny Edge Detection. Metode ini efektif untuk mendeteksi garis lurus dalam gambar berdasarkan parameter jarak (ρ) dan sudut (θ). Hasil simulasi menampilkan gambar asli di sebelah kiri dan garis yang terdeteksi di sebelah kanan. Garis yang berhasil dideteksi ditampilkan dengan warna hijau. Keakuratan deteksi garis dipengaruhi oleh parameter threshold pada Canny dan resolusi Hough Transform. Hasil ini menunjukkan kemampuan OpenCV dalam menemukan fitur garis dalam gambar yang memiliki tepi tajam.

2. Template Matching untuk Deteksi Objek

Simulasi ini menggunakan metode Template Matching untuk mendeteksi keberadaan objek dalam gambar berdasarkan template yang diberikan. Hasilnya menampilkan posisi objek dengan kotak hijau yang ditandai pada gambar utama. Metode ini menggunakan fungsi ``cv2.matchTemplate`` dan mencari lokasi maksimum dengan ``cv2.minMaxLoc``. Template yang digunakan harus lebih kecil dari gambar utama agar metode ini efektif. Kesuksesan deteksi objek dipengaruhi oleh kesamaan antara template dan bagian gambar utama, serta pencahayaan yang seragam.

3. Pembuatan Pyramid Gambar

Pada simulasi ini, dilakukan pembuatan Gaussian Pyramid untuk mengecilkan resolusi gambar dalam beberapa level. Gambar asli berada pada Level 0, dan setiap level berikutnya dihasilkan menggunakan fungsi ``cv2.pyrDown``, yang mengurangi ukuran gambar sebesar setengahnya. Hasil simulasi menampilkan gambar pada empat level dengan resolusi yang semakin berkurang di setiap tingkat. Pyramid gambar ini berguna dalam berbagai aplikasi, seperti deteksi fitur multi-skala, kompresi gambar, dan analisis visual yang efisien.

4. Deteksi Lingkaran Menggunakan Hough Transform

Simulasi ini bertujuan untuk mendeteksi bentuk lingkaran dalam gambar menggunakan metode Hough Circle Transform. Setelah gambar dikonversi ke grayscale dan melalui proses deteksi tepi, lingkaran-lingkaran yang terdeteksi ditampilkan dengan garis hijau. Pusat lingkaran juga ditandai dengan titik merah. Hasil simulasi menunjukkan bahwa parameter seperti threshold, jarak antar lingkaran, dan ukuran radius mempengaruhi jumlah lingkaran yang terdeteksi. Jika parameter tidak diatur dengan benar, dapat terjadi deteksi ganda atau tidak ada lingkaran yang terdeteksi sama sekali.

5. Ekstraksi Warna Dominan Menggunakan K-Means Clustering

Pada simulasi ini, dilakukan ekstraksi warna dominan dari gambar menggunakan metode K-Means Clustering. Gambar dikonversi menjadi array piksel, kemudian dikelompokkan ke dalam beberapa cluster warna. Hasil simulasi menampilkan gambar asli beserta warna dominan dalam kotak terpisah. Warna-warna dominan ini dihitung berdasarkan pusat cluster dari algoritma K-Means. Simulasi ini menunjukkan bahwa metode ini efektif untuk menemukan warna utama dalam gambar dengan cepat, dan sering digunakan dalam aplikasi seperti kompresi gambar dan analisis visual.

6. Deteksi Kontur pada Gambar

Simulasi ini bertujuan untuk mendeteksi kontur objek dalam gambar setelah melalui proses Gaussian Blur dan Adaptive Thresholding untuk mengurangi noise dan meningkatkan kontras. Kontur yang terdeteksi digambar di atas gambar asli menggunakan fungsi `cv2.findContours`. Hasil simulasi menampilkan kontur dalam warna hijau di gambar hasil. Kontur ini berguna untuk mendeteksi batas objek dan fitur penting dalam gambar. Parameter seperti ukuran area kontur mempengaruhi hasil akhir, memastikan hanya kontur besar yang terdeteksi.

7. Simulasi Lidar Data Extraction and Obstacle Detection di Webots

Simulasi terakhir menggunakan robot yang dilengkapi LIDAR dan sensor jarak untuk mendeteksi rintangan dan menghindarinya secara otomatis. LIDAR mengumpulkan data jarak dalam bentuk range image, sementara sensor jarak (ultrasonik) digunakan untuk menentukan jarak ke objek di sisi kiri dan kanan robot. Kecepatan roda kiri dan kanan dihitung berdasarkan data dari sensor, sehingga robot dapat menyesuaikan arah gerakannya. Hasil simulasi menunjukkan robot mampu menghindari rintangan dengan efektif berkat kombinasi data sensor LIDAR dan algoritma pengendali sederhana.