

Nama : Fadhil Dzikri Aqila

NIM : 1103213136

Kelas : TK-45-G09

Analisis Simulasi Image Processing dan Feature Detection dengan Python dan OpenCV

1. Filter Moving Average

Filter moving average adalah salah satu teknik smoothing sederhana yang menghitung rata-rata nilai intensitas piksel dalam kernel persegi panjang. Ini digunakan untuk mengurangi noise seragam, menghasilkan gambar yang lebih halus dengan detail lebih sedikit. Dalam simulasi, filter diterapkan dengan fungsi `cv2.filter2D()`. Dengan kernel ukuran $k \times k$, filter ini membuat gambar tampak buram, tetapi dapat menghilangkan noise tingkat rendah. Kelemahan utama adalah hilangnya ketajaman pada tepi objek, yang dapat memengaruhi analisis lanjutan seperti deteksi fitur.

2. Deteksi Fitur dengan SIFT (Scale-Invariant Feature Transform)

SIFT digunakan untuk mendeteksi dan mendeskripsikan fitur dalam gambar secara robust terhadap perubahan skala, rotasi, dan pencahayaan. Dalam simulasi, OpenCV menyediakan fungsi `cv2.SIFT_create()` untuk mendeteksi keypoints dan menghasilkan deskriptor fitur. Ketika diterapkan pada gambar kompleks seperti pemandangan atau objek bertekstur, SIFT mampu mengidentifikasi fitur penting seperti sudut, ujung tekstur, dan pola yang unik. Hasil simulasi menunjukkan SIFT sangat andal dalam tugas seperti pencocokan objek pada gambar yang dirotasi atau diskalakan.

3. Representasi Histogram Gambar

Histogram digunakan untuk menganalisis distribusi piksel dalam gambar. Dengan `cv2.calcHist()`, kita dapat membuat histogram untuk saluran warna tertentu atau gambar grayscale. Representasi histogram memberikan informasi tentang karakteristik visual gambar, seperti sebaran kecerahan atau dominasi warna tertentu. Dalam aplikasi praktis, histogram digunakan untuk kontras enhancement, seperti equalization, yang membuat gambar lebih jelas. Simulasi ini menunjukkan pentingnya histogram dalam pemrosesan awal atau analisis lebih lanjut.

4. Gaussian Smoothing

Gaussian smoothing menggunakan kernel berbobot Gaussian untuk mengurangi noise sambil mempertahankan detail tepi lebih baik dibandingkan filter moving average. Implementasi dengan `cv2.GaussianBlur()` pada simulasi memperlihatkan efek smoothing yang lebih alami karena bobot kernel yang berkurang seiring dengan jarak dari pusat. Simulasi ini menunjukkan manfaat Gaussian smoothing dalam preprocessing, terutama untuk mempersiapkan gambar sebelum deteksi fitur atau segmentasi objek.

5. Deteksi Tepi dengan Sobel Filter

Sobel filter adalah metode deteksi tepi berbasis gradien. Dengan menggunakan kernel khusus untuk menghitung perubahan intensitas dalam arah xxx dan yyy, Sobel filter dapat mendeteksi tepi horizontal, vertikal, atau kombinasi keduanya. Dalam simulasi, fungsi `cv2.Sobel()` menghasilkan gambar dengan tepi yang menonjol. Filter ini penting untuk segmentasi atau analisis bentuk, tetapi rentan terhadap noise jika gambar tidak dihaluskan sebelumnya.

6. Representasi Fitur dengan HOG (Histogram of Oriented Gradients)

HOG adalah metode deskripsi fitur yang memanfaatkan distribusi gradien orientasi. Ini sangat efektif untuk deteksi objek seperti manusia atau kendaraan. Dengan OpenCV, HOG dihitung menggunakan gradien orientasi setiap sel piksel dan dirangkum dalam histogram. Simulasi ini menunjukkan bahwa HOG sangat sensitif terhadap pola lokal pada gambar, menjadikannya ideal untuk pengenalan objek yang berbasis tekstur atau bentuk.

Analisis Simulasi Robot dengan Webots dan OpenCV

1. Visual Tracking dengan OpenCV

Visual tracking adalah teknik pelacakan objek secara real-time. Dalam simulasi, bola merah dilacak dengan mengubah gambar ke ruang warna HSV menggunakan `cv2.cvtColor()` dan menerapkan thresholding untuk menyorot warna tertentu. Mask biner yang dihasilkan menunjukkan posisi objek dalam frame. Posisi centroid dihitung dengan `cv2.moments()`, dan pengendali P sederhana digunakan untuk mengatur pergerakan robot berdasarkan selisih posisi bola dari titik tengah frame. Kelebihan pendekatan ini adalah kesederhanaannya, namun sensitivitas terhadap pencahayaan atau warna latar belakang dapat menjadi kendala.

2. Document Scanner Simulation

Simulasi ini melibatkan beberapa langkah penting dalam image processing:

- **Preprocessing:** Thresholding adaptif dengan `cv2.adaptiveThreshold()` untuk membuat gambar biner dan menonjolkan dokumen.
- **Deteksi Kontur:** `cv2.findContours()` digunakan untuk mendeteksi kontur dokumen, yang diidentifikasi sebagai persegi panjang terbesar dalam frame.
- **Transformasi Perspektif:** Menggunakan `cv2.getPerspectiveTransform()` untuk melakukan warp transform sehingga dokumen tampil dalam perspektif top-down. Simulasi ini menyerupai aplikasi scanner ponsel dan sangat berguna untuk mendigitalisasi dokumen secara efisien.

Namun, terdapat error pada project nya yaitu tekstur box atau kardus yang besar tidak ter-load dengan benar sehingga Document Scanner tidak berfungsi untuk box yang besar, tetapi berfungsi untuk box yang kecil.

3. Fruit Detection Robot

Simulasi deteksi buah melibatkan kombinasi computer vision dan kontrol robotik.

- **Deteksi Buah:** Warna buah dianalisis dalam ruang warna HSV. Thresholding diterapkan untuk setiap warna buah tertentu, seperti merah untuk apel atau kuning untuk pisang. Kontur buah diekstrak dengan `cv2.findContours()`.
- **Lokalisasi:** Koordinat centroid dihitung, memberikan posisi untuk kontrol robot.
- **Pengendalian Lengan Robot:** Koordinat digunakan untuk mengarahkan lengan robot secara presisi ke lokasi buah dan memindahkannya. Dalam simulasi ini, OpenCV berfungsi sebagai alat utama untuk pengenalan objek, sedangkan pengontrol robot menangani manipulasi fisik.