

1. Image Processing dalam Webots

Pemrosesan gambar dalam simulasi Webots adalah langkah awal yang digunakan untuk memahami data visual yang dihasilkan oleh kamera robot. Data ini biasanya dalam format mentah atau sebagai buffer gambar, yang memerlukan pengolahan untuk menjadi informasi yang bermakna.

Langkah-langkah Utama:

1. Mengambil Gambar:

- Dalam Webots, gambar diambil dari kamera robot menggunakan metode seperti:

```
python  
camera.getImage()
```

Gambar ini biasanya berupa buffer dalam format mentah yang perlu dikonversi.

2. Konversi Ruang Warna:

- Gambar sering diubah ke format grayscale atau HSV menggunakan OpenCV:
 - **Grayscale:** Menyederhanakan data visual untuk analisis tepi atau kontur.
 - **HSV:** Mempermudah segmentasi warna dengan membagi warna (Hue), intensitas (Saturation), dan nilai (Value). Contoh kode:

```
python  
img_hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

3. Segmentasi Warna:

- Untuk mendeteksi objek tertentu (seperti bola merah atau buah), digunakan nilai rentang HSV:

```
python  
mask = cv2.inRange(img_hsv, lower_bound, upper_bound)
```

Mask ini menghasilkan area biner: piksel dalam rentang warna bernilai 1, lainnya bernilai 0.

4. Penghapusan Noise:

- Dilakukan dengan filter seperti Gaussian atau median blur untuk membersihkan citra.

```
python  
blurred = cv2.GaussianBlur(mask, (5, 5), 0)
```

Contoh dari Repositori:

- **webots-example-visual-tracking:** Bola merah dideteksi dengan segmentasi warna pada ruang HSV, yang memisahkan bola dari latar belakang dengan mudah.
 - **Fruit-Detection-Robot-with-OpenCV-and-Webots:** Warna buah diisolasi menggunakan segmentasi warna serupa.
-

2. Feature Detection dalam Webots

Feature Detection adalah langkah untuk mengenali pola atau elemen visual yang signifikan dalam gambar, seperti tepi, sudut, atau bentuk.

Langkah-langkah Utama:

1. Deteksi Tepi:

- Algoritma seperti Canny digunakan untuk mendeteksi tepi dalam gambar:

```
python
edges = cv2.Canny( grayscale_image, threshold1, threshold2)
```

- **Threshold rendah:** Menentukan sensitivitas untuk mendeteksi tepi tipis.
- **Threshold tinggi:** Menentukan sensitivitas untuk tepi yang lebih jelas.

2. Deteksi Sudut:

- Digunakan metode seperti Harris Corner untuk mendeteksi sudut objek:

```
python
corners = cv2.cornerHarris( grayscale_image, block_size, ksize, k)
```

3. Deteksi Kontur:

- Kontur adalah kurva yang menghubungkan piksel dengan intensitas tertentu. Digunakan untuk mengenali bentuk objek.

```
python
contours, _ = cv2.findContours( mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

Contoh dari Repositori:

- **document_scanner_simulation:**
 - Gambar dari kamera robot diproses untuk mendeteksi tepi dokumen menggunakan Canny.
 - Kontur terbesar diidentifikasi sebagai dokumen.
- **Fruit-Detection-Robot-with-OpenCV-and-Webots:**
 - Kontur buah dikenali dari gambar setelah segmentasi warna.

3. Feature Description dalam Webots

Feature Description adalah proses menghasilkan representasi unik dari fitur yang telah terdeteksi untuk memudahkan pencocokan atau pengenalan.

Langkah-langkah Utama:

1. Ekstraksi Deskriptor:

- Algoritma seperti ORB atau SIFT digunakan untuk menghasilkan deskriptor vektor:

```
python  
keypoints, descriptors = orb.detectAndCompute(image, None)
```

2. Pencocokan Fitur:

- Fitur yang dideskripsikan digunakan untuk mencocokkan objek antar gambar dengan cv2.BFMatcher:

```
Python  
matcher = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)  
matches = matcher.match(descriptor1, descriptor2)
```

3. Penyaringan Outlier:

- Untuk meningkatkan akurasi pencocokan, outlier disaring menggunakan teknik seperti RANSAC.

Contoh dari Repositori:

- **document_scanner_simulation:**
 - Fitur dokumen dideskripsikan untuk memperkirakan transformasi perspektif, meratakan tampilan dokumen dalam gambar.
- **Fruit-Detection-Robot-with-OpenCV-and-Webots:**
 - Deskripsi fitur digunakan untuk membedakan buah berdasarkan bentuk atau warna.