

ANALISIS TUGAS WEEK 9

1. Camera robot untuk deteksi blob warna

- Inisialisasi Robot dan Perangkat

Pada bagian awal, robot dan perangkat-perangkat utamanya seperti kamera dan motor diinisialisasi:

- Kamera diaktifkan dengan `wb_camera_enable()`, dan ukuran gambar kamera disimpan dalam variabel `width` dan `height`.
- Motor kiri dan kanan diatur pada posisi tak terbatas (`INFINITY`), memungkinkan kontrol langsung terhadap kecepatannya.

Kelemahan:

- Tidak ada validasi apakah perangkat berhasil diakses.
- Tidak ada opsi konfigurasi fleksibel, seperti mengubah resolusi kamera atau menyesuaikan parameter motor.

- Logika Utama dalam Loop

○ Deteksi Warna Blob:

Kode memproses area tertentu dari gambar untuk mendeteksi warna blob dominan (merah, hijau, atau biru):

```
for (i = width / 3; i < 2 * width / 3; i++) {  
    for (j = height / 2; j < 3 * height / 4; j++) {  
        red += wb_camera_image_get_red(image, width,  
            i, j);  
        green += wb_camera_image_get_green(image,  
            width, i, j);  
        blue += wb_camera_image_get_blue(image,  
            width, i, j);  
    }  
}
```

Analisis:

Area gambar yang diproses hanya bagian tengah gambar diperiksa ($\frac{1}{3}$ lebar dan $\frac{1}{4}$ tinggi dari bawah). Ini membantu mengurangi noise dari tepi gambar.

Logika perbandingan intensitas warna: Blob warna dianggap terdeteksi jika satu komponen RGB jauh lebih besar dari dua lainnya (rasio 3:1).

Kekuatan:

- Efisien: Memproses sebagian kecil gambar mempercepat deteksi.
- Toleransi noise dasar: Rasio intensitas 3:1 mengurangi deteksi warna palsu.

Kelemahan:

- Sensitivitas terhadap pencahayaan: Intensitas RGB dapat bervariasi tergantung pencahayaan.
- Blob kompleks tidak didukung: Tidak dapat mendeteksi warna campuran atau gradasi.
- Area terbatas: Blob di luar pusat gambar tidak terdeteksi.

○ Respons Robot terhadap Blob

Robot memiliki tiga kondisi utama yang diatur oleh variabel `pause_counter`:

- Blob Baru Terdeteksi: Robot berhenti sejenak, menyimpan gambar, dan menunggu sebelum melanjutkan pencarian. Nama file gambar berdasarkan warna blob.
- Blob Lama Masih Dekat: Robot mulai berputar untuk menghindari mendeteksi blob yang sama.
- Tidak Ada Blob: Robot terus berputar untuk mencari blob baru.

Kelebihan:

- Respons sederhana dan efektif: Robot berhenti untuk memastikan gambar tersimpan dengan baik.
- Rotasi terus-menerus: Meningkatkan peluang menemukan blob.

Kekurangan:

- Rotasi tanpa arah: Robot tidak diarahkan ke lokasi blob yang terdeteksi.
- Blob sama dapat dideteksi berulang jika robot tidak bergerak cukup jauh.
- Gerakan tidak optimal: Tidak ada manuver maju-mundur untuk mendekati blob.

- Penyimpanan Gambar

Gambar blob yang terdeteksi disimpan di direktori pengguna dengan nama sesuai warna:

```
char *filepath;
#ifdef _WIN32
const char *user_directory =
wbu_system_short_path(wbu_system_getenv("USERPROFILE"));
#else
const char *user_directory = wbu_system_getenv("HOME");
#endif
strcat(filepath, filenames[current_blob]);
wb_camera_save_image(camera, filepath, 100);
```

Kelebihan:

- Gambar dapat diverifikasi secara manual.
- Resolusi 100% memastikan kualitas gambar terjaga.

Kekurangan:

- Nama file hardcoded menyebabkan file sebelumnya tertimpa jika warna yang sama terdeteksi lagi.
- Tidak ada penamaan dinamis untuk membedakan gambar berdasarkan waktu atau lokasi.

2. Camera robot dengan fokus berdasarkan objek yang ada di depannya

- Inisialisasi Robot dan Perangkat

- Kamera:
Kamera diaktifkan dengan `wb_camera_enable()`, memungkinkan pengambilan data visual.
- Sensor Jarak:
Sensor jarak diaktifkan menggunakan `wb_distance_sensor_enable()` untuk mengukur jarak objek di depan robot.
- Motor:
Dua motor roda (kiri dan kanan) diatur dalam mode kecepatan (posisi tak terbatas INFINITY) dengan kecepatan awal untuk rotasi di tempat.

- Loop Utama

Dalam loop utama, robot membaca data sensor jarak dan menyesuaikan fokus kamera sesuai jarak objek.

- Pengukuran Jarak Objek

```
const double object_distance =  
wb_distance_sensor_get_value(distance_sensor) / 1000;
```

Nilai dari sensor jarak diubah ke satuan meter (dengan membagi 1000, asumsi satuan asli adalah milimeter).

Analisis:

Nilai sensor dikalibrasi untuk keperluan fokus kamera. Ini penting karena fungsi `wb_camera_set_focal_distance()` menerima parameter dalam satuan meter.

- Penyesuaian Fokus Kamera

```
wb_camera_set_focal_distance(camera,  
object_distance);
```

Jarak fokus kamera diatur agar sesuai dengan jarak objek. Jika tidak ada objek (jarak terlalu jauh), fokus kamera akan diatur ke jarak maksimum sensor.

Kelebihan:

- Respons adaptif: Kamera menyesuaikan fokus secara real-time sesuai jarak objek.
- Kemampuan dinamis: Fokus otomatis mempermudah menangkap gambar dengan jelas meskipun objek bergerak.

Kekurangan:

- Jarak minimum sensor: Jika objek terlalu dekat (di bawah jarak minimum sensor), fokus kamera mungkin tidak akurat.
- Respons linear sederhana: Tidak ada penyesuaian non-linear jika respons fokus kamera tidak proporsional dengan jarak.

- Rotasi Robot

```
wb_motor_set_velocity(left_motor, -SPEED);  
wb_motor_set_velocity(right_motor, SPEED);
```

Robot berputar di tempat, memungkinkan kamera memindai area sekitarnya untuk mendeteksi objek.

Kelebihan:

- Penyapuan area: Rotasi di tempat memungkinkan kamera menjelajah lingkungan.
- Pencarian objek sederhana: Rotasi memastikan objek dalam radius kamera terdeteksi.

Kekurangan:

- Rotasi tanpa arah: Tidak ada logika untuk mengarahkan robot ke objek tertentu.

3. Camera robot deteksi blob berwarna pada robot dengan efek *motion blur camera*

- Inisialisasi Perangkat

Kode memulai dengan menginisialisasi perangkat-perangkat utama:

- Kamera:

Kamera diaktifkan dengan `wb_camera_enable()` untuk menangkap gambar. Resolusi kamera diperoleh melalui fungsi `wb_camera_get_width()` dan `wb_camera_get_height()`.

- Motor:

Dua motor (roda kiri dan kanan) diatur dalam mode kecepatan dengan posisi tak terbatas menggunakan `wb_motor_set_position()`. Kecepatan awal motor disiapkan dengan nilai nol.
- Konstanta Warna:

Warna yang didukung adalah merah, hijau, dan biru. Setiap warna memiliki nama (string) dan kode warna ANSI untuk mencetak ke terminal.
- Deteksi Blob Berwarna

Robot mendeteksi warna blob di tengah bidang pandang kamera dengan pendekatan berbasis pixel averaging.

- Analisis Gambar Kamera

```
for (i = width / 3; i < 2 * width / 3; i++) {
    for (j = height / 2; j < 3 * height / 4; j++) {
        red += wb_camera_image_get_red(image, width, i, j);
        blue += wb_camera_image_get_blue(image, width, i, j);
        green += wb_camera_image_get_green(image, width, i,
j);
    }
}
```

Wilayah gambar yang diperiksa dibatasi di tengah layar. Nilai rata-rata dari komponen warna merah, hijau, dan biru dihitung.

- Penentuan Blob

```
if ((red > 3 * green) && (red > 3 * blue))
    current_blob = RED;
else if ((green > 3 * red) && (green > 3 * blue))
    current_blob = GREEN;
else if ((blue > 3 * red) && (blue > 3 * green))
    current_blob = BLUE;
else
    current_blob = NONE;
```

Blob diidentifikasi jika satu komponen warna jauh lebih dominan dibandingkan yang lain (dengan faktor pengali 3).

Kekuatan:

- Efisiensi ruang pencarian: Hanya bagian tengah gambar yang diperiksa, mengurangi kebutuhan komputasi.

Kelemahan:

- Tidak tahan terhadap noise atau variasi warna akibat efek motion blur.
- Blob kecil di tepi gambar mungkin tidak terdeteksi.

- Logika Pergerakan Robot

Robot memiliki tiga skenario utama untuk bergerak berdasarkan status deteksi blob:

- Blob Tidak Ditemukan: Robot berputar di tempat untuk mencari blob baru. Kecepatan motor kiri dan kanan memiliki arah berlawanan.
- Blob Baru Terdeteksi: Robot berhenti, menyimpan gambar dengan blob, dan menunggu beberapa saat (pause counter).
- Menunggu Sebelum Mencari Blob Baru: Robot mengabaikan gambar untuk sementara waktu, mencegah pendeteksian ulang blob yang sama.

Kekuatan:

- Responsif terhadap blob baru: Robot berhenti untuk fokus pada blob yang terdeteksi.
- Pause counter: Mencegah deteksi ulang yang tidak diinginkan.

Kelemahan:

- Tidak ada gerakan maju/mundur: Robot hanya berputar di tempat tanpa kemampuan mendekati blob.
- Pause counter tetap: Tidak fleksibel terhadap variasi lingkungan atau kecepatan rotasi.

- Efek Motion Blur

Efek motion blur terjadi ketika kamera menangkap gambar dari objek yang bergerak cepat, menghasilkan gambar yang kabur. Hal ini meliputi:

- Blob yang Tidak Jelas: Perpaduan warna dapat membuat blob sulit diidentifikasi. Kriteria perbandingan warna (faktor 3x) mungkin tidak tercapai akibat smearing.
- Kesalahan Deteksi: Warna blob bisa salah diidentifikasi karena nilai rata-rata warna menjadi ambigu. Blob kecil bisa terabaikan jika terlalu kabur.
- Penurunan Akurasi Lokasi: Wilayah blob yang terdeteksi mungkin bergeser karena kaburnya batas tepi blob.

4. Robot dengan kamera: deteksi blob berwarna dengan *noise mask*

- Inisialisasi Kamera dan Motor:
 - Kamera menangkap gambar untuk analisis warna.
 - Motor dikendalikan untuk mengatur gerakan robot sesuai deteksi blob.
- Analisis Gambar Kamera:
 - Robot membaca nilai warna dari bagian tengah gambar (sepertiga horizontal dan setengah vertikal).
 - Warna dominan diidentifikasi menggunakan logika perbandingan intensitas warna.
- Logika Pergerakan:
 - Jika blob tidak terdeteksi, robot berputar untuk mencari.
 - Jika blob terdeteksi, robot berhenti dan menyimpan gambar blob.
- Dampak Noise Mask pada Kinerja

Noise mask adalah gangguan acak pada gambar yang dapat memengaruhi deteksi warna blob. Dampaknya:

 - Kesalahan Identifikasi Warna: Noise dapat mengubah nilai intensitas piksel, sehingga warna dominan sulit diidentifikasi dengan logika sederhana.
 - Blob Tidak Terdeteksi: Jika noise mendominasi wilayah target, nilai warna dapat menyebar sehingga tidak ada warna yang melebihi ambang batas.
 - Pengaruh pada Pergerakan Robot: Dengan tidak adanya deteksi blob, robot akan terus berputar tanpa mencapai blob.
 - Hasil Gambar yang Terganggu: Gambar yang disimpan dapat menunjukkan hasil yang kabur atau terdistorsi.

5. Deteksi objek dengan kamera dan pengenalan objek pada robot

- Inisialisasi Robot dan Perangkat
 - Kamera: Mengaktifkan kamera dengan fungsi `wb_camera_enable`. Mengaktifkan fitur pengenalan objek menggunakan `wb_camera_recognition_enable`.

- Motor: Memungkinkan kontrol kecepatan dengan INFINITY pada target posisi.
 - Deteksi dan Pengenalan Objek
 - Mendapatkan jumlah objek: `wb_camera_recognition_get_number_of_objects` menghitung jumlah objek yang dikenali kamera.
 - Informasi objek: `wb_camera_recognition_get_objects` mengembalikan data detail, termasuk:
 - Model dan ID objek.
 - Posisi relatif dalam ruang 3D.
 - Orientasi relatif dalam quaternion.
 - Ukuran objek di dunia nyata dan di citra kamera.
 - Warna objek.
 - Kontrol Pergerakan Robot
 - Robot berputar dengan kecepatan tetap untuk memindai lingkungan.
 - Output

Kode menghasilkan output seperti ini:

Jumlah objek dikenali: Recognized 1 objects.

Informasi detail objek:

Model of object 0: jam jar

Id of object 0: 179

Relative position of object 0: 0.369144 -0.118401 -0.021659

Relative orientation of object 0: -0.010775 0.010119 0.999891 2.555273

Size of object 0: 0.093218 0.090000

Position of the object 0 on the camera image: 204 79

Size of the object 0 on the camera image: 73 81

Analisis warna objek: Menyebutkan warna dominan objek, seperti:

 - Color 1/2: 0.550000 0.060000 0.060000
 - Color 2/2: 0.860000 0.880000 0.900000
6. Implementasi segmentasi kamera pada robot menggunakan webots
- Inisialisasi Perangkat
 - Kamera:
 - Aktifkan kamera: `wb_camera_enable`.
 - Aktifkan pengenalan objek: `wb_camera_recognition_enable`.
 - Aktifkan segmentasi objek: `wb_camera_recognition_enable_segmentation`.
 - Segmentasi memungkinkan pemetaan warna tertentu untuk objek yang dikenali.
 - Display:
 - Menampilkan hasil segmentasi menggunakan perangkat tampilan virtual (`wb_display`).
 - Motor:
 - Dikendalikan untuk menggerakkan robot dengan kecepatan konstan (mode rotasi).
 - Segmentasi dan Tampilan
 - Memeriksa segmentasi: `wb_camera_recognition_is_segmentation_enabled` mengecek apakah segmentasi aktif.

- Mengambil citra segmentasi:
wb_camera_recognition_get_segmentation_image mengembalikan data citra segmentasi.
 - Menampilkan citra:
Fungsi wb_display_image_paste digunakan untuk menampilkan citra segmentasi.
 - Proses yang Terjadi
 - Pengambilan Data Kamera:
Kamera menangkap citra asli dan melakukan segmentasi untuk memisahkan objek.
 - Citra Segmentasi:
Kamera menghasilkan citra segmentasi berupa gambar dalam format BGRA:
B: Blue.
G: Green.
R: Red.
A: Alpha (transparansi).
 - Pergerakan Robot:
Robot berputar terus-menerus selama segmentasi berlangsung.
 - Output yang Dihasilkan
 - Citra Segmentasi: Gambar dengan warna yang dipisahkan berdasarkan fitur segmentasi.
 - Tampilan Display: Hasil segmentasi divisualisasikan di jendela simulasi.
7. Implementasi penggunaan kamera bola pada robot menggunakan webots
- Inisialisasi Robot dan Kamera:
Fungsi wb_robot_init() digunakan untuk menginisialisasi robot simulasi. Kamera diaktifkan menggunakan wb_camera_enable() untuk menangkap gambar. Resolusi gambar diambil dari fungsi wb_camera_get_width() dan wb_camera_get_height().
 - Sensor Jarak Ultrasonik:
Sensor ultrasonik digunakan untuk mendeteksi jarak ke objek di sekitar, dengan dua sensor ditempatkan di sisi kiri dan kanan robot.
 - Blob Warna (Segmen Objek):
Fungsi wb_camera_get_image() mendapatkan gambar dari kamera bola. Gambar dipindai pixel demi pixel untuk mendeteksi warna berdasarkan nilai RGB (merah, hijau, biru).
 - Blob Deteksi:
Pixel yang memenuhi threshold untuk masing-masing warna disimpan koordinatnya (color_index).
 - Konversi Koordinat ke Sudut:
Fungsi coord2D_to_angle() digunakan untuk mengubah koordinat 2D dari kamera menjadi sudut dalam koordinat polar, memudahkan penentuan arah objek relatif terhadap robot.
 - Pengendalian Robot:
Nilai dari sensor jarak digunakan untuk menghitung kecepatan roda dengan koefisien tertentu (coefficients). Kecepatan roda kiri dan kanan diatur menggunakan fungsi wb_motor_set_velocity().
 - Tampilan dan Output:

Posisi dan sudut dari blob yang terdeteksi (warna merah, hijau, dan biru) dicetak ke terminal menggunakan `printf()`. Hasil keluaran memberikan informasi lokasi blob terakhir yang terlihat dalam bentuk koordinat dan sudut.