

Nama : Hamdan Syaifuddin Zuhri

NIM : 1103220220

Kelas : TK-45-G09

Analisis Tugas Robotika Week 9

Kamera pada robot di Webots dapat digunakan untuk berbagai tugas seperti navigasi, deteksi objek, dan penghindaran hambatan. Tugas utama dari kamera adalah untuk menyediakan data visual yang dapat digunakan oleh robot untuk mengambil keputusan secara real-time.

1. World Camera

- Fungsi: Kamera ini digunakan untuk mendeteksi dan mengenali objek berwarna (blob) yang ada di depan robot. Dalam kode ini, robot menggunakan kamera untuk mencari objek berwarna merah, hijau, atau biru, dan berhenti jika menemukan salah satu dari objek tersebut.
- Hasil: Robot dapat mendeteksi objek berdasarkan warna yang ada di layar kamera dan berhenti untuk menyimpan gambar objek yang ditemukan.
- Algoritma:
 - Kode ini menggunakan kamera untuk menangkap gambar dan menganalisis warna piksel di area tengah gambar.
 - Jika warna merah lebih dominan dibandingkan hijau dan biru, maka dianggap sebagai objek merah. Proses yang sama diterapkan untuk warna hijau dan biru.
 - Ketika objek ditemukan, robot akan berhenti dan menyimpan gambar dari objek tersebut.

- Contoh pada code:

```
camera = wb_robot_get_device("camera");
wb_camera_enable(camera, time_step);
const unsigned char *image = wb_camera_get_image(camera);
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);

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;
    }

```

2. World Camera Auto Focus

- Fungsi: Kamera ini memiliki kemampuan auto-focus yang memungkinkan kamera untuk secara otomatis menyesuaikan fokus berdasarkan jarak objek yang terdeteksi oleh sensor jarak.
- Hasil: Kamera akan menyesuaikan jarak fokusnya agar objek yang lebih dekat atau lebih jauh selalu terlihat jelas di gambar.
- Algoritma:
 - Sensor jarak digunakan untuk mendeteksi kedekatan objek dengan robot.
 - Berdasarkan jarak objek, fungsi `wb_camera_set_focal_distance()` digunakan untuk menyesuaikan fokus kamera secara otomatis agar objek yang lebih dekat terlihat jelas.

- Contoh pada code:

```

camera = wb_robot_get_device("camera");
wb_camera_enable(camera, TIME_STEP);
distance_sensor = wb_robot_get_device("distance sensor");
wb_distance_sensor_enable(distance_sensor, TIME_STEP);

const double object_distance = wb_distance_sensor_get_value(distance_sensor) / 1000;
wb_camera_set_focal_distance(camera, object_distance);

```

3. World Camera Motion Blur

- Fungsi: Kamera ini digunakan untuk mensimulasikan efek motion blur, di mana objek yang bergerak cepat akan terlihat buram dalam gambar.
- Hasil: Gambar yang dihasilkan akan menunjukkan efek blur pada objek yang bergerak cepat, yang memperlihatkan arah gerakan objek tersebut.
- Algoritma:

- Kode ini menggunakan logika untuk mencari "blob" warna yang berbeda di area gambar dan menggerakkan robot sesuai dengan posisi objek tersebut.
- Efek motion blur terjadi ketika robot bergerak dengan kecepatan tinggi dan objek di depan kamera terlihat buram.

- Contoh pada code:

```
camera = wb_robot_get_device("camera");
wb_camera_enable(camera, time_step);
const unsigned char *image = wb_camera_get_image(camera);

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);
    }
}
```

4. World Camera Noise Mask

- Fungsi: Kamera ini mensimulasikan efek noise pada gambar, yang digunakan untuk menguji bagaimana robot mengolah gambar yang terdistorsi.
- Hasil: Gambar akan mengandung noise atau gangguan acak yang disisipkan, yang mengganggu pengenalan objek.
- Algoritma: Noise acak ditambahkan pada setiap piksel gambar untuk mensimulasikan gangguan visual yang mempengaruhi pengolahan citra.

- Contoh pada code:

```
addNoiseToImage(image);
```

5. World Camera Recognition

- Fungsi: Kamera ini memiliki kemampuan untuk mengenali objek tertentu di dalam gambar dengan menggunakan teknologi pengenalan objek.
- Hasil: Kamera dapat mengenali dan menampilkan informasi objek yang terdeteksi, seperti model, ID, posisi relatif, dan warna objek.

- Algoritma: Kamera menggunakan kemampuan pengenalan objek untuk mendeteksi objek dalam gambar dan menampilkan berbagai informasi terkait objek yang dikenali, termasuk posisi dan ukuran.

- Contoh pada code:

```
camera = wb_robot_get_device("camera");
wb_camera_enable(camera, TIME_STEP);
wb_camera_recognition_enable(camera, TIME_STEP);

int number_of_objects = wb_camera_recognition_get_number_of_objects(camera);
const WbCameraRecognitionObject *objects =
wb_camera_recognition_get_objects(camera);
for (i = 0; i < number_of_objects; ++i) {
    printf("Model of object %d: %s\n", i, objects[i].model);
    // Menampilkan informasi objek lainnya
}
```

6. World Camera Segmentation

- Fungsi: Kamera ini menggunakan kemampuan segmentasi untuk membagi gambar menjadi beberapa bagian berdasarkan warna atau fitur lain.
- Hasil: Gambar akan terbagi menjadi beberapa segmen, memungkinkan robot untuk mengenali bagian-bagian gambar yang berbeda.
- Algoritma: Kamera melakukan segmentasi gambar untuk membagi objek atau area berdasarkan kriteria yang telah ditentukan, seperti warna atau kedalaman.

- Contoh pada code:

```
camera = wb_robot_get_device("camera");
wb_camera_enable(camera, TIME_STEP);
wb_camera_recognition_enable(camera, TIME_STEP);
wb_camera_recognition_enable_segmentation(camera);

const unsigned char *data = wb_camera_recognition_get_segmentation_image(camera);
if (data) {
    segmented_image = wb_display_image_new(display, width, height, data,
WB_IMAGE_BGRA);
```

```

wb_display_image_paste(display, segmented_image, 0, 0, false);
wb_display_image_delete(display, segmented_image);
}

```

7. World Camera

- Fungsi: Kamera ini memberikan pandangan 360 derajat dari lingkungan sekitar robot, memungkinkan robot untuk mendapatkan pandangan penuh tentang lingkungan sekitarnya.
- Hasil: Gambar 360 derajat dihasilkan, yang memungkinkan robot untuk melihat seluruh lingkungan sekitarnya.
- Algoritma: Gambar 360 derajat dikumpulkan dan diproses untuk menampilkan gambar lengkap dari seluruh lingkungan. Data sensor jarak juga digunakan untuk mengendalikan kecepatan motor robot berdasarkan jarak ke objek di sekitar robot.

- Contoh pada code:

```

camera = wb_robot_get_device("camera");
wb_camera_enable(camera, 2 * TIME_STEP);
int width = wb_camera_get_width(camera);
int height = wb_camera_get_height(camera);

const unsigned char *image = wb_camera_get_image(camera);
for (x = 0; x < width; x++) {
    for (y = 0; y < height; y++) {
        r = wb_camera_image_get_red(image, width, x, y);
        g = wb_camera_image_get_green(image, width, x, y);
        b = wb_camera_image_get_blue(image, width, x, y);
        if (r > THRESHOLD && g < THRESHOLD && b < THRESHOLD) {
            color_index[RED][X] = x;
            color_index[RED][Y] = y;
        }
    }
}
}

```

Setiap jenis kamera yang digunakan dalam simulasi Webots memiliki aplikasi yang berbeda. World Camera digunakan untuk mendeteksi warna, Camera_Auto_Focus untuk penyesuaian fokus otomatis, Camera_Motion_Blur untuk efek gerakan, Camera_Noise_Mask untuk menguji ketahanan

terhadap gangguan, Camera_Recognition untuk pengenalan objek, Camera_Segmentation untuk segmentasi gambar, dan Spherical_Camera untuk pandangan 360 derajat dari lingkungan robot. Setiap kamera memanfaatkan algoritma pengolahan citra yang sesuai dengan tujuan penggunaannya dalam simulasi.

