

Analisis Tugas 9 Robotika

Camera Robot untuk Deteksi Blob Warna

Program ini memanfaatkan **kamera** untuk mendeteksi **warna tertentu** (merah, hijau, atau biru) dan mengontrol gerakan robot berdasarkan deteksi warna tersebut. Proses utama dimulai dengan inisialisasi perangkat kamera serta motor kiri dan kanan, yang diatur dalam mode kontrol kecepatan. Kamera diaktifkan untuk menangkap gambar dengan resolusi yang disesuaikan. Lalu, program memasuki loop utama, di mana setiap siklus waktu langkah (`time_step`) program mengambil gambar baru dan menganalisis piksel di bagian tengah layar untuk mendeteksi keberadaan blob (bintik warna) dengan memisahkan komponen merah, hijau, dan biru.

Berdasarkan hasil analisis warna, robot dapat mengambil berbagai tindakan. Jika sebuah blob warna ditemukan, robot akan berhenti, menyimpan gambar tersebut ke direktori pengguna, dan masuk ke mode "pause". Jika tidak ada blob yang terdeteksi, robot terus berputar sambil menganalisis gambar baru. Kode ini juga menggunakan fungsi ANSI untuk memberikan output teks berwarna di terminal, menunjukkan warna blob yang terdeteksi. Program ini mengontrol robot secara langsung dengan kecepatan motor kiri dan kanan menggunakan fungsi dari API Webots seperti `wb_motor_set_velocity()` dan `wb_camera_save_image()`, serta mengelola siklus hidup program dengan `wb_robot_step()` dan `wb_robot_cleanup()`.

Camera Robot Dengan Fokus Kamera Berdasarkan Objek yang Ada Didepannya

Program ini memanfaatkan **kamera** dan **sensor jarak** untuk mengatur **fokus kamera** secara dinamis berdasarkan jarak objek yang terdeteksi. Awalnya, program menginisialisasi robot dan mengaktifkan perangkat keras seperti kamera dan sensor jarak dengan interval waktu (`TIME_STEP`) sebesar 32 milidetik. Kamera dan sensor jarak diaktifkan menggunakan fungsi `wb_camera_enable()` dan `wb_distance_sensor_enable()`, sementara motor kiri dan kanan diatur dalam mode kontrol kecepatan dengan fungsi `wb_motor_set_position()` yang disetel ke **INFINITY**, sehingga memungkinkan robot bergerak bebas. Robot diinstruksikan untuk berputar dengan kecepatan motor kiri dan kanan yang berlawanan.

Dalam **loop utama**, program terus menerus membaca nilai jarak dari sensor menggunakan `wb_distance_sensor_get_value()`, mengonversinya ke satuan meter, dan menyesuaikan **fokus kamera** dengan jarak tersebut melalui `wb_camera_set_focal_distance()`. Dengan cara ini, kamera robot secara otomatis menyesuaikan fokusnya untuk memastikan objek di berbagai jarak tetap terlihat jelas. Program terus berjalan hingga simulasi dihentikan, dan pada akhir siklus hidupnya, fungsi `wb_robot_cleanup()` dipanggil untuk membersihkan sumber daya sebelum keluar dari program. Pendekatan ini berguna untuk simulasi robot yang membutuhkan respons visual dinamis terhadap objek di sekitarnya.

Camera Robot Deteksi Blob berwarna pada Robot dengan Efek Motion Blur Camera

Program dimulai dengan inisialisasi perangkat keras robot, termasuk kamera dan motor roda kiri serta kanan. Kamera diaktifkan untuk menangkap gambar pada interval waktu tertentu, sementara motor diatur untuk kontrol kecepatan. Program ini memanfaatkan fungsi `wb_camera_get_image()` untuk mengambil data gambar dan memeriksa piksel di bagian tengah gambar guna mendeteksi intensitas warna merah, hijau, dan biru. Berdasarkan perbandingan nilai warna, robot menentukan apakah blob merah, hijau, atau biru terdeteksi.

Dalam loop utama, robot mengambil tindakan sesuai kondisi blob. Jika tidak ada blob yang terdeteksi, robot terus berputar untuk mencari blob baru. Jika blob ditemukan, robot berhenti, mencetak warna blob yang terdeteksi ke terminal menggunakan warna teks ANSI, dan menyimpan gambar blob tersebut ke direktori pengguna. File gambar disimpan dengan nama sesuai warna blob yang terdeteksi, seperti **red_blob.png** untuk blob merah. Setelah menyimpan gambar, robot memasuki mode jeda (`pause_counter`), mencegah deteksi ulang blob yang sama. Program ini memberikan contoh bagaimana robot dapat berinteraksi secara visual dengan lingkungannya dan menyesuaikan perilaku berdasarkan pengolahan citra waktu nyata.

Deteksi Blob berwarna dengan Noise Mask

Program dimulai dengan inisialisasi perangkat kamera dan motor robot. Kamera diaktifkan untuk menangkap gambar setiap siklus waktu (`time_step`), dan robot memproses bagian tengah gambar untuk menganalisis intensitas warna merah, hijau, dan biru menggunakan fungsi seperti `wb_camera_image_get_red()`. Nilai-nilai ini kemudian digunakan untuk menentukan apakah terdapat blob berwarna merah, hijau, atau biru di dalam gambar berdasarkan perbandingan intensitas warna tersebut.

Dalam **loop utama**, robot mengambil tindakan sesuai kondisi yang terdeteksi. Jika blob terdeteksi, robot berhenti, mencetak warna blob ke terminal dengan kode warna ANSI, dan menyimpan gambar ke direktori pengguna dengan nama yang sesuai seperti **red_blob.png**. Jika tidak ada blob yang terdeteksi, robot berputar untuk mencari blob baru. Setelah menyimpan gambar, robot memasuki mode jeda (`pause_counter`), mencegah deteksi ulang blob yang sama selama beberapa waktu. Program ini mengilustrasikan bagaimana robot dapat berinteraksi dengan lingkungannya secara visual dan mengubah perilaku berdasarkan analisis citra waktu nyata.

Deteksi Objek dengan Camera dan Pengenalan Objek pada Robot

Program dimulai dengan inisialisasi perangkat kamera dan motor. Kamera diaktifkan untuk menangkap gambar secara periodik dengan interval waktu (`TIME_STEP`) sebesar 64 milidetik, serta fitur pengenalan objek diaktifkan menggunakan `wb_camera_recognition_enable()`. Motor robot dikontrol dalam mode kecepatan menggunakan fungsi `wb_motor_set_velocity()` untuk mengatur gerakan berputar.

Dalam **loop utama**, program terus memeriksa jumlah objek yang dikenali oleh kamera menggunakan `wb_camera_recognition_get_number_of_objects()` dan mencetak jumlah tersebut ke terminal. Untuk setiap objek yang dikenali, program menampilkan berbagai informasi, seperti

model, ID, posisi relatif, orientasi, ukuran, posisi pada gambar kamera, serta warna objek. Informasi ini berguna untuk pemrosesan visual yang lebih lanjut, seperti navigasi berbasis objek atau interaksi robot dengan lingkungannya. Program ini menampilkan bagaimana robot dapat secara dinamis mengenali dan menganalisis objek di sekitarnya dalam simulasi waktu nyata.

Segmentasi Camera pada Robot

Program dimulai dengan inisialisasi kamera, display, dan motor robot. Kamera diaktifkan menggunakan `wb_camera_enable()` dan fitur pengenalan serta segmentasi diaktifkan dengan `wb_camera_recognition_enable()` dan `wb_camera_recognition_enable_segmentation()`. Informasi dimensi kamera (lebar dan tinggi) diambil menggunakan `wb_camera_get_width()` dan `wb_camera_get_height()`. Selain itu, perangkat **display** digunakan untuk menampilkan hasil segmentasi secara visual.

Dalam **loop utama**, robot memeriksa apakah segmentasi telah diaktifkan dan apakah kamera aktif dalam mengambil sampel. Jika ya, data gambar segmentasi diambil menggunakan `wb_camera_recognition_get_segmentation_image()`. Data gambar ini kemudian diubah menjadi objek **WbImageRef** dan ditampilkan di layar menggunakan fungsi **Display API**, seperti `wb_display_image_new()` dan `wb_display_image_paste()`. Setelah gambar ditampilkan, memori gambar dibebaskan menggunakan `wb_display_image_delete()`. Dengan ini, program dapat memvisualisasikan hasil segmentasi secara real-time, yang berguna untuk navigasi robot atau tugas pengenalan lingkungan yang kompleks.

Kamera Bola pada Robot

Program dimulai dengan inisialisasi perangkat keras robot, termasuk kamera, sensor jarak, dan motor roda. Kamera diaktifkan untuk menangkap gambar pada interval waktu yang ditentukan (`TIME_STEP`), dan data warna diekstrak menggunakan fungsi `wb_camera_image_get_red()`, `wb_camera_image_get_green()`, serta `wb_camera_image_get_blue()`. Sensor jarak digunakan untuk menghitung kecepatan motor berdasarkan jarak terdeteksi, dengan koefisien tertentu untuk menentukan arah pergerakan robot.

Dalam **loop utama**, program membaca nilai dari sensor jarak dan gambar dari kamera untuk menganalisis posisi blob warna dalam koordinat gambar. Jika blob warna melebihi ambang batas (`THRESHOLD`), posisi blob akan dicatat. Posisi ini kemudian digunakan untuk menghitung sudut blob relatif terhadap robot menggunakan fungsi `coord2D_to_angle()`. Informasi ini ditampilkan ke terminal, memberi tahu pengguna posisi terakhir dari blob dan sudutnya. Motor robot kemudian disesuaikan berdasarkan nilai kecepatan yang dihitung, memungkinkan robot bergerak menuju blob yang terdeteksi, memberikan simulasi navigasi visual berbasis objek secara real-time.