

LAPORAN UJIAN AKHIR SEMESTER ROBOTIKA

“Penggunaan Callback Function untuk Analisis Data Sensor dan Pengolahan Gambar dalam ROS”

Ditujukan Untuk Memenuhi Ujian Akhir Semester Robotika



Disusun Oleh :

Muhammad Fharist (110320202)

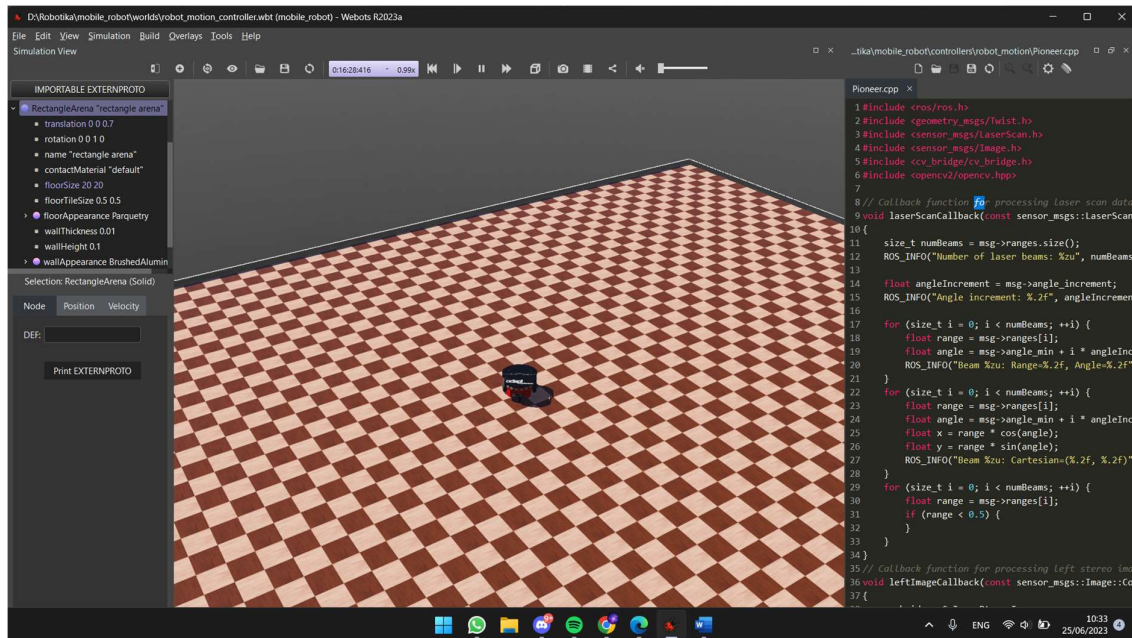
PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2023

TECHNICAL REPORT



```
#include <ros/ros.h>

#include <geometry_msgs/Twist.h>

#include <sensor_msgs/LaserScan.h>

#include <sensor_msgs/Image.h>

#include <cv_bridge/cv_bridge.h>

#include <opencv2/opencv.hpp>

// Callback function for processing laser scan data
void laserScanCallback(const sensor_msgs::LaserScan::ConstPtr& msg)
{
    size_t numBeams = msg->ranges.size();

    ROS_INFO("Number of laser beams: %zu", numBeams);

    float angleIncrement = msg->angle_increment;

    ROS_INFO("Angle increment: %.2f", angleIncrement);
```

```

for (size_t i = 0; i < numBeams; ++i) {
    float range = msg->ranges[i];
    float angle = msg->angle_min + i * angleIncrement;
    ROS_INFO("Beam %zu: Range=%.2f, Angle=%.2f", i, range, angle);
}

for (size_t i = 0; i < numBeams; ++i) {
    float range = msg->ranges[i];
    float angle = msg->angle_min + i * angleIncrement;
    float x = range * cos(angle);
    float y = range * sin(angle);
    ROS_INFO("Beam %zu: Cartesian=(%.2f, %.2f)", i, x, y);
}

for (size_t i = 0; i < numBeams; ++i) {
    float range = msg->ranges[i];
    if (range < 0.5) {
    }
}
}

// Callback function for processing left stereo image data
void leftImageCallback(const sensor_msgs::Image::ConstPtr& msg)
{
    cv_bridge::CvImagePtr cvImage;
    try {
        cvImage = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
    } catch (cv_bridge::Exception& e) {
        ROS_ERROR("CV Bridge Exception: %s", e.what());
        return;
    }

    cv::Mat processedImage;
    cv::cvtColor(cvImage->image, processedImage, cv::COLOR_BGR2GRAY);
    cv::imshow("Left Image", processedImage);
    cv::waitKey(1);
}

```

```

// Callback function for processing right stereo image data
void rightImageCallback(const sensor_msgs::Image::ConstPtr& msg)
{
    cv_bridge::CvImagePtr cvImage;
    try {
        cvImage = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
    } catch (cv_bridge::Exception& e) {
        ROS_ERROR("CV Bridge Exception: %s", e.what());
        return;
    }
    cv::Mat processedImage;
    cv::cvtColor(cvImage->image, processedImage, cv::COLOR_BGR2GRAY);
    cv::imshow("Right Image", processedImage);
    cv::waitKey(1);
}

int main(int argc, char** argv)
{
    ros::init(argc, argv, "pioneer_controller");
    ros::NodeHandle nh;
    ros::Subscriber laserSub = nh.subscribe("laser_scan", 10, laserScanCallback);
    ros::Subscriber leftImageSub = nh.subscribe("left_image", 10, leftImageCallback);
    ros::Subscriber rightImageSub = nh.subscribe("right_image", 10, rightImageCallback);
    ros::Publisher twistPub = nh.advertise<geometry_msgs::Twist>("cmd_vel", 10);
    while (ros::ok()) {
        geometry_msgs::Twist twistMsg;
        // Set linear and angular velocities
        twistMsg.linear.x = 0.5; // Contoh kecepatan linear maju 0.5 m/s
        twistMsg.angular.z = 0.2; // Contoh kecepatan angular sebesar 0.2 rad/s
        twistPub.publish(twistMsg);

        ros::spinOnce();
    }
}

```

```
loop_rate.sleep();  
}  
return 0;  
}
```

PENDAHULUAN

Kodingan ini merupakan contoh program yang digunakan dalam lingkungan ROS (Robot Operating System) untuk mengendalikan robot Pioneer. Program ini menggunakan beberapa topik ROS untuk menerima data sensor dan mengirimkan perintah gerakan ke robot.

Program ini terdiri dari tiga callback function yang dipanggil saat data sensor diterima, yaitu laserScanCallback, leftImageCallback, dan rightImageCallback. Fungsi laserScanCallback digunakan untuk memproses data pemindaian laser, sedangkan fungsi leftImageCallback dan rightImageCallback digunakan untuk memproses data gambar stereo.

Selain itu, terdapat juga fungsi utama main yang menginisialisasi node ROS, membuat subscriber untuk menerima data sensor, membuat publisher untuk mengirimkan perintah gerakan, dan menjalankan loop utama yang mengontrol gerakan robot.

DESKRIPSI

Berikut merupakan deskripsi tiap fungsi yang terdapat pada kodingan tersebut :

- Callback Function ‘laserScanCallback’

Fungsi laserScanCallback pada kodingan ini adalah callback function yang dipanggil saat data pemindaian laser diterima. Fungsi ini melakukan pemrosesan data pemindaian laser, termasuk menampilkan informasi jumlah beam laser, inkremental sudut, jarak setiap beam, dan mengubah koordinat polar menjadi koordinat kartesian.

1. Mendapatkan informasi jumlah beam laser:

- Variabel numBeams diinisialisasi dengan ukuran msg->ranges, yang merupakan vektor yang berisi jarak setiap beam.
- Informasi jumlah beam laser dicetak menggunakan ROS_INFO.

2. Mendapatkan inkremental sudut:

- Variabel angleIncrement diinisialisasi dengan msg->angle_increment, yang merupakan inkremental sudut antara setiap beam.
- Informasi inkremental sudut dicetak menggunakan ROS_INFO.

3. Memproses data jarak dan sudut setiap beam:

- Melakukan iterasi untuk setiap beam menggunakan loop for.

- Dalam loop, variabel range diinisialisasi dengan jarak beam ke-i menggunakan `msg->ranges[i]`.
 - Variabel angle diinisialisasi dengan sudut beam ke-i menggunakan `msg->angle_min + i * angleIncrement`.
 - Informasi jarak dan sudut setiap beam dicetak menggunakan `ROS_INFO`.
4. Mengubah koordinat polar menjadi koordinat kartesian:
 - Melakukan iterasi untuk setiap beam menggunakan loop for.
 - Dalam loop, variabel range diinisialisasi dengan jarak beam ke-i menggunakan `msg->ranges[i]`.
 - Variabel angle diinisialisasi dengan sudut beam ke-i menggunakan `msg->angle_min + i * angleIncrement`.
 - Variabel x diinisialisasi dengan nilai `range * cos(angle)`, yang merupakan koordinat x dalam sistem kartesian.
 - Variabel y diinisialisasi dengan nilai `range * sin(angle)`, yang merupakan koordinat y dalam sistem kartesian.
 - Informasi koordinat kartesian setiap beam dicetak menggunakan `ROS_INFO`
 5. Pemrosesan tambahan (tidak ada implementasi):
 - Terdapat loop for tambahan yang tidak memiliki implementasi. Loop ini dapat digunakan untuk melakukan pemrosesan tambahan sesuai dengan kebutuhan.
- Callback Function 'leftImageCallback'
- Fungsi `leftImageCallback` adalah callback function yang dipanggil saat data gambar stereo kiri diterima. Fungsi ini melakukan pemrosesan data gambar, khususnya mengubah gambar berwarna menjadi gambar grayscale menggunakan library OpenCV.
1. Konversi gambar berwarna ke grayscale:
 - Variabel `cvImage` merupakan pointer ke objek `cv_bridge::CvImage`, yang mengonversi pesan gambar ROS menjadi citra OpenCV.
 - Melakukan konversi gambar berwarna (BGR) menjadi gambar grayscale menggunakan `cv::cvtColor` dengan parameter `cv::COLOR_BGR2GRAY`.
 - Citra hasil konversi disimpan dalam variabel `processedImage`.
 2. Menampilkan gambar:
 - Citra grayscale ditampilkan menggunakan `cv::imshow` dengan judul jendela "Left Image".
 - `cv::waitKey(1)` digunakan untuk memberikan waktu tampilan gambar.
- Callback Function 'rightImageCallback'

Fungsi `rightImageCallback` adalah callback function yang dipanggil saat data gambar stereo kanan diterima. Fungsi ini melakukan pemrosesan data gambar, khususnya mengubah gambar berwarna menjadi gambar grayscale menggunakan library OpenCV.

1. Konversi gambar berwarna ke grayscale:

- Variabel `cvImage` merupakan pointer ke objek `cv_bridge::CvImage`, yang mengonversi pesan gambar ROS menjadi citra OpenCV.
- Melakukan konversi gambar berwarna (BGR) menjadi gambar grayscale menggunakan `cv::cvtColor` dengan parameter `cv::COLOR_BGR2GRAY`.
- Citra hasil konversi disimpan dalam variabel `processedImage`.

2. Menampilkan gambar:

- Citra grayscale ditampilkan menggunakan `cv::imshow` dengan judul jendela "Right Image".
- `cv::waitKey(1)` digunakan untuk memberikan waktu tampilan gambar.

- Main Loop

Bagian main loop bertanggung jawab untuk mengendalikan robot berdasarkan data yang diterima dan mempublikasikan perintah gerakan melalui topik `cmd_vel`. Dalam loop ini, dilakukan beberapa operasi seperti:

- Membuat objek `geometry_msgs::Twist` untuk mengatur kecepatan linear dan angular robot.
- Mengatur nilai kecepatan linear dan angular sesuai kebutuhan (dalam contoh ini, `linear.x = 0.5` dan `angular.z = 0.2`).
- Mem-publish perintah gerakan menggunakan objek `twistPub`.

KESIMPULAN

Pada kodingan ini, terdapat tiga callback function yang berfungsi untuk memproses data sensor dalam lingkungan ROS. Fungsi `laserScanCallback` memproses data pemindaian laser, menghitung jumlah beam laser, inkremental sudut, dan mengubah koordinat polar menjadi koordinat kartesian. Fungsi `leftImageCallback` dan `rightImageCallback` memproses data gambar stereo kiri dan kanan, mengkonversi gambar berwarna menjadi grayscale menggunakan library OpenCV.

Dengan menggunakan callback function ini, pengguna dapat memproses dan menganalisis data sensor secara real-time, serta melakukan operasi pemrosesan lanjutan sesuai dengan kebutuhan aplikasi. Selain itu, dengan menampilkan gambar menggunakan OpenCV, pengguna dapat memvisualisasikan data gambar secara interaktif.

Pemahaman dan modifikasi terhadap callback function ini memungkinkan pengguna untuk mengembangkan aplikasi yang lebih kompleks dalam pengembangan robot menggunakan ROS, seperti deteksi objek, pemetaan lingkungan, atau navigasi berdasarkan visi.