

Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

Отчёт

по лабораторной работе №1

Дисциплина: Техническое зрение

Тема: Моделирование движения робота с использованием библиотеки OpenCV

Студент гр. 3331506/70401

Кондратченко О.О.

Преподаватель

Варлашин В.В.

« » _____ 2020 г.

Санкт-Петербург

2020

Задание

С помощью методов *OpenCV* реализовать движение робота по заданному полю, его поворот на месте, а также ограничение на выезд за пределы.

Ход работы

Общий алгоритм выполнения программы:

1. Ожидание нажатия кнопок управления;
2. Вычисление координат точек робота, если нажата какая-то кнопка продольного или поперечного движения;
3. Вычисление координат точек робота, если нажата какая-то из кнопок поворота;
4. Проверка на пересечение роботом границ заданного поля;
5. Вывод нового изображения робота на экран.

Весь процесс находится в цикле *while(1)*, пока не будет нажата клавиша «*esc*».

Для того, чтобы на экране не отображалось предыдущее изображение робота, создан отдельный объект класса *Mat* – общий фон. Главное изображение становится клоном общего фона, таким образом стирается предыдущее изображение робота, а затем уже на пустом экране рисуется следующее положение робота.

Класс *myRobot*

Для создания робота был создан класс *myRobot*. Параметры данного класса представлены на рисунке 1.

```
private:
// Позиционирование
// Координаты центра
Point2f m_center;
// Угол поворота
float m_angle;
// Область движения
Size2i m_area;

//Размеры робота
// Ширина корпуса
float m_width;
// Высота корпуса
float m_height;
// Ширина колеса
float m_wheelWidth;
// Радиус колеса
float m_wheelRadius;

//Линейная скорость и угловая
float m_speed;
float m_angularSpeed;
float m_amplitude;
```

Рисунок 1 – Параметры класса

Основные функции

1) Функция линейного перемещения *move()*

Данная функция осуществляет перемещение центра робота. При этом вычисляются 4 крайние точки центра, которые проверяем на пересечение границ. В случае пересечения одной из четырех сторон – выполняется обратное действие (т.е отмена линейного движения).

Так как начало системы координат изображения находится в верхнем левом углу, при движении вперед необходимо уменьшать у-координату центра робота. Расчет центра робота по оси X,Y представлен на рисунке 2.

```
m_center.x += m_amplitude *(directionX * cos(m_angle) - directionY * sin(m_angle));
m_center.y += m_amplitude *(directionX * sin(m_angle) + directionY * cos(m_angle));
```

Рисунок 2 – Расчет центра робота при линейном движении

2) Функция углового вращения *rotate()*

Аналогично с функцией *move()* производит поворот, при этом снова по 4 точкам корпуса робота производится проверка на выход изображения за границы. В случае выхода действие “отменяется” (т.е производится обратное действие).

3) Функция отрисовки робота *draw()*

Модель робота с колесами представляет из себя 16 линий. В функции *draw()* вычисляются координаты точек каждой линии в глобальной системе координат. Привязка точек к форме робота представлено на рисунке 3.

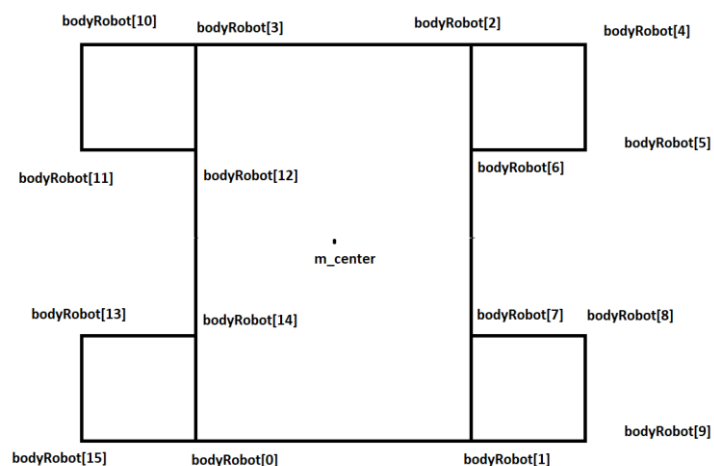


Рисунок 3 – Привязка точек к корпусу робота

На рисунке 4 показан пример расчета корпуса робота.

```
// Корпус робота
bodyRobot[0].x = m_center.x - m_width / 2 * cos(m_angle) - m_height / 2 * sin(m_angle);
bodyRobot[0].y = m_center.y + m_height / 2 * cos(m_angle) - m_width / 2 * sin(m_angle);

bodyRobot[1].x = m_center.x + m_width / 2 * cos(m_angle) - m_height / 2 * sin(m_angle);
bodyRobot[1].y = m_center.y + m_height / 2 * cos(m_angle) + m_width / 2 * sin(m_angle);

bodyRobot[2].x = m_center.x + m_width / 2 * cos(m_angle) + m_height / 2 * sin(m_angle);
bodyRobot[2].y = m_center.y - m_height / 2 * cos(m_angle) + m_width / 2 * sin(m_angle);

bodyRobot[3].x = m_center.x - m_width / 2 * cos(m_angle) + m_height / 2 * sin(m_angle);
bodyRobot[3].y = m_center.y - m_height / 2 * cos(m_angle) - m_width / 2 * sin(m_angle);
```

Рисунок 4 – расчет координатов точек корпуса

Отрисовка всех линий содержится так же в этой функции. с помощью встроенной в OpenCV функции `cv::line()`.

Вывод

В ходе работы проведено успешное моделирование движения робота по заданной области. Успешно изучены и использованы необходимые алгоритмы *OpenCV*.