

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

Водорезов Г.И.

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()*, 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. Робот состоит из 20 линий, заданных 16 точками. Координаты точек рассчитываются по формуле 1.

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) & t_x \\ \sin(\varphi) & \cos(\varphi) & t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

Вывод всех линий на экран производится с помощью встроенной в *OpenCV* функции *cv::line()*.

Класс *MyRobot*

Для создания робота был создан класс *MyRobot*. Данный класс обладает параметрами, показанными на рисунке 1.

```
//позиционирование
Point2f m_center;           //координаты центра
Point2f m_point[20];        //точки робота
float m_angle;               //угол наклона
Size2i m_area;               //рабочая область робота

//размеры
float m_width;               //ширина робота
float m_height;              //длина робота
float m_wheelWidth;          //ширина колес
float m_wheelRadius;         //радиус колес

//скорости
float m_speed;               //линейная скорость
float m_angularSpeed;        //угловая скорость
```

Рисунок 1 – Параметры класса

Координаты центра устанавливаются с помощью функции *setCenter(Mat image)*, которая в начальный момент времени устанавливает центр робота в центр заданного поля.

Область движения устанавливается с помощью функции *setArea(Size2i area)*, которая областью движения робота выбирает область заданными размерами.

Все остальные параметры устанавливаются при вызове конструктора.

Основные методы

1) Метод *move()*

Данный метод принимает 2 параметра – направление движения по осям X и Y. Далее происходит расчет нового центра в зависимости от принимаемых параметров, проверка пересечения границ, вызов метода для расчета всех точек робота.

2) Метод *rotate()*

Вычисление нового угла поворота робота осуществлено в функции *rotate()*. Аналогично с методом *move()* происходит проверка на то, разрешено ли повернуться в заданном направлении или нет, затем вычисляется новый угол поворота робота.

Данный метод принимает единственный параметр – направление поворота робота.

3) Метод *pointCalculate()*

При вызове данного метода происходит расчет координат для всех 20 точек робота. Не имеет входных параметров.

4) Метод *draw()*

Данный метод принимает на вход объект типа *Mat* – исходный фон. На данном фоне происходит отрисовка всех 20 линий и возврат исходного фона с отрисованным на нем роботом.

Вывод

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