

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

Жернаков А.А.

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.

## Задачи

- 1) Реализовать вывод изображения робота на экран;
- 2) Реализовать продольное и поперечное движения робота;
- 3) Реализовать поворот робота относительно центра его корпуса;
- 4) Реализовать ограничения на выезд за пределы заданной области.

## 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.

## Класс *MyRobot*

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

```
Point2f m_center; // координаты центра
float m_angle; // угол поворота
float m_width; // ширина корпуса
float m_height; // высота корпуса
float m_wheelWidth; // ширина колеса
float m_wheelDiameter; // диаметр колеса
float m_speed; // скорость движения
float m_angularSpeed; // угловая скорость
Size2i m_area; // область движения
```

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(Mat image)*, которая областью движения робота выбирает созданный объект класса *Mat*.

Все остальные параметры устанавливаются при вызове конструктора.

## Основные функции

### 1) Функция *move()*

Сканирование нажатия клавиш продольного и поперечного движений и одновременно с этим вычисление новых координат центра робота осуществлено в функции *Move()*.

При нажатии какой-либо из клавиш происходит проверка на то, разрешено ли двигаться в этом направлении или нет, затем вычисляются новые координаты для центра робота.

Пример с нажатием клавиши «a» показан на рисунке 2.

```

char pressedKeyMove = waitKey(0);
if (pressedKeyMove == 'a' || pressedKeyMove == 'A')
{
    if (isA)
    {
        m_center.x = m_center.x - m_speed * cos(m_angle);
        m_center.y = m_center.y - m_speed * sin(m_angle);
    }
}

```

Рисунок 2 – Пример обработки движения

Одной из особенностей функции *move()* является то, что начало системы координат изображения находится в верхнем левом углу, поэтому при движении вперед(клавиша «w») необходимо уменьшать у-координату центра робота.

## 2) Функция *rotate()*

Сканирование нажатия клавиш поворота и одновременно с этим вычисление нового угла поворота робота осуществлено в функции *rotate()*.

Аналогично с функцией *move()* происходит проверка на то, разрешено ли повернуться в этом направлении или нет, затем вычисляется новый угол поворота робота.

Пример показан на рисунке 3.

```

char pressedKeyRotate = waitKey(0);
if (pressedKeyRotate == 'e' || pressedKeyRotate == 'E')
{
    if(isE)
        m_angle = m_angle + m_angularSpeed * pi / 180;
}

```

Рисунок 3 – Пример обработки поворота

## 3) Функция *draw()*

Проверка на пересечение роботом границ заданного поля, а также вывод нового изображения робота на экран производится функцией *draw()*.

Модель робота с колесами представляет из себя 16 линий. Изначально в функции *draw()* вычисляются координаты точек каждой линии в глобальной системе координат. Для этого используется вспомогательная функция *calculationNewCoord()*. Данная функция выполняет произведение матрицы поворота на матрицу координат точки в локальной системе координат робота, как показано в формуле 1.

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

Затем происходит вывод всех линий на экран с помощью функции *line()*.

Проверка на пересечение роботом границ заданного поля происходит следующим образом:

- 1) Определение, какую границу пересек робот;
- 2) Определение положения робота при пересечении данной границы;
- 3) Блокировка движения в направлении границы.

### **Вывод**

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