

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

Коновалов В.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.

## Ход работы

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, приведёнными рисунке 1.

```
float m_speed;  
float m_angularSpeed;  
Point2f m_center;  
Size2i m_area;  
float m_angle;  
float m_width;  
float m_height;  
float m_wheelWidth;  
float m_wheelDiameter;
```

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()*

Вычисление новых координат центра робота осуществлено в функции *Move()*.

При нажатии какой-либо из клавиш (проверка нажатия происходит в отдельном файле *realization.cpp*) вычисляются новые координаты для центра робота.

Пример с изменением координат при движении влево (нажатии клавиши «a») показан на рисунке 2.

```
...  
  
    // влево  
case 2:  
    m_center.x = m_center.x - m_speed * cos(m_angle);  
    m_center.y = m_center.y - m_speed * sin(m_angle);  
  
    break;  
  
...
```

Рисунок 2 – Пример обработки движения

### 2) Функция *rotate()*

Вычисление нового угла поворота робота осуществлено в функции *rotate()*.

Аналогично с функцией *move()* вычисляется новый угол поворота робота.

Пример показан на рисунке 3.

```
// для поворота по часовой
case 1:
    m_angle = m_angle - m_angularSpeed * pi / 180;
    break;

    ...
```

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 путём произведения матрицы поворота на матрицу координат точки в локальной системе координат робота, как показано в формуле 1.

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

Проверка на пересечение роботом границ заданного поля происходит следующим образом:

- 1) Каждый раз сохраняем изначальное положение робота;
- 2) После передвижения проверяем, пересекли ли крайние точки колёс робота границы;
- 3) Если пересекли, то приравниваем новое положение изначально сохранённому (то есть по сути не даём роботу двигаться дальше).

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

## **Вывод**

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