

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

Чернов Е.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 кнопка движения или поворота;
- 3) Проверка на пересечение роботом границ заданного поля;
- 5) Вывод нового изображения робота на экран.

Весь процесс находится в цикле *while()*.

Для того, чтобы на экране не отображалось предыдущее изображение робота, создан отдельный объект класса *Mat* – общий фон. Главное изображение становится копией 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*.

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.

Робот описывается с помощью центра, ширины, высоты и диаметром и шириной колес. Координаты ключевых точек в локальной системе

координат, по которым будет обрисовываться робот хранятся и инициализируются с помощью `std::vector<cv::Point> m_localCoordinates`.

Основные методы

1) Метод *play* ()

Сканирование нажатия клавиш и одновременно с этим вычисление смещения центра и угла поворота робота реализовано в методе *play()*. Ожидание нажатия клавиш осуществлено с помощью конструкции *switch case*.

Вычисление смещения необходимо для проверки на выезд за пределы границ роботом. Если же ни одна ключевая точка робота не пересекла границы поля, то центр изменяется с учетом данных смещений, в противном случае будет отрисован робот с несмещённым центром.

Пример с нажатием клавиши “w” показан на рисунке 1.

```
case (char)'w':
    deltaCenterX = -m_speed * cos(m_angle + M_PI / 2);
    deltaCenterY = -m_speed * sin(m_angle + M_PI / 2);
    flag = checkEdge(deltaCenterX, deltaCenterY, 0);
    if (flag == -1)
    {
        break;
    }
    imageDelete.copyTo(ioimage);
    draw(ioimage, deltaCenterX, deltaCenterY, 0);
    m_center.x = m_center.x + deltaCenterX;
    m_center.y = m_center.y + deltaCenterY;
    cv::imshow("robot", ioimage);
    break;
```

Рисунок 1 – Пример обработки движения

Для проверки на пересечение границ роботом используется вспомогательный метод *checkEdge()*. При этом проверяются только точки углов робота на пересечение с границей. При пересечении движение робота блокируется.

Одной из особенностей метода *play()* 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) Метод *draw()*

Отображение линий между точками реализовано в методе *draw()*.

Модель робота с колесами представляет из себя 12 линий. Изначально в методе *draw()* вычисляются координаты точек каждой линии в глобальной системе координат. Для этого используется вспомогательный метод *convertToWorldCoordinates()*. Данный метод выполняет произведение матрицы поворота на матрицу координат точки в локальной системе координат робота, как показано в формуле 1.

$$\begin{pmatrix} X_W \\ Y_W \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \phi & -\sin \phi & t_x \\ \sin \phi & \cos \phi & t_y \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} X_L \\ Y_L \\ 1 \end{pmatrix} \quad (1)$$

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

Вывод

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