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 Засецкий В.С.

Преподаватель Варлашин В.В.

«7» 10 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 скорости робота, если нажата одна из кнопок линейного перемещения.

3) Вычисление угловой скорости робота, если нажата одна из кнопок поворота.

4) Вычисление координат точек робота.

5) Проверка на пересечение роботом границ заданного поля.

6) 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 описания робота был создан класс MyRobot. Его атрибуты приведены на рисунке 1.

```
struct Speed
{
    float x;
    float y;
};

Point2f m_center; //координаты центра
float m_angle; // угол поворота локальной СК относительно глобальной
float m_width; //ширина корпуса
float m_height; //высота корпуса
float m_wheelWidth; //ширина колёс
float m_wheelDiameter; //диаметр колёс

Speed m_speed; //скорость робота (в локальной СК по осям X и Y)
float m_angularSpeed; //угловая скорость

Size2i m_area; //область движения
```

Рисунок 1 — Атрибуты класса MyRobot

Следует отметить, что в объекте Mat OpenCV начало отсчёта находится в левом верхнем углу, при этом ось X направлена вправо, а ось Y — вниз. Начало координат локальной системы координат робота находится в его центре, а оси направлены аналогично: X — вправо, Y — вниз. Иллюстрация приведена на рисунке 2.

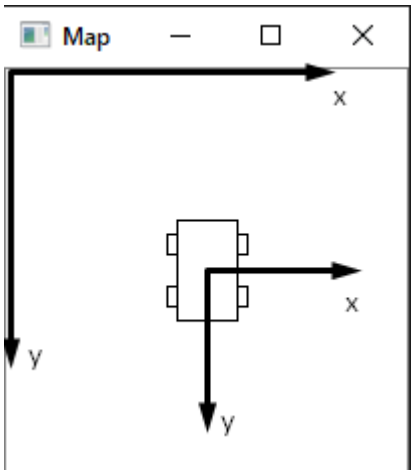

Рисунок 2 — Расположение систем координат

Функции-сеттеры класса приведены на рисунке 3.

```
void setSpeed(const Speed speed);
void setAngularSpeed(const float angularSpeed);
void setArea(const Size2i area);
int setArea(Mat image);
int setCenter(Mat image);
int setCenter(float x, float y);
```

Рисунок 3 — Функции-сеттеры класса MyRobot

Функции *setArea()* и *setCenter()* перегружены: могут принимать значения напрямую или подстраиваться под изображение.

Функции-геттеры приведены на рисунке 4.

```
Speed getSpeed();
float getAngularSpeed();
```

Рисунок 4 — Функции-геттеры класса MyRobot

Атрибуты *width, height, angle, wheelWidth, wheelDiameter, speed, angularSpeed* можно задать при вызове конструктора класса.

Основные функции

1. Функция *move()*

В функции вычисляется новое значение координаты центра робота в глобальной системе координат.

2. Функция *rotate()*

В функции выполняется вычисляется новое значение угловой скорости.

3. Функция *draw()*

В функции производится вычисление глобальных координат точек робота, отрисовывается изображение робота, а также выполняется проверка на пересечение роботом границ изображения. Если граница пересечена, робот тормозится (линейная и угловая скорость принимают значение 0), а на изображение выводится текст, сообщающий о пересечении границ.

Вычисление глобальных координат точек робота производится во вспомогательной функции *setCoordinates()* по следующим формулам:

$$x_{\text{гл}} = x_c + x_{\text{л}} \cdot \cos\alpha + y_{\text{л}} \cdot \sin\alpha,$$

$$y_{\text{гл}} = y_c + y_{\text{л}} \cdot \cos\alpha - x_{\text{л}} \cdot \sin\alpha,$$

где x_c, y_c — глобальные координаты центра робота,

$x_{\text{л}}$, $y_{\text{л}}$ — локальные координаты точки робота,

α — угол поворота глобальной системы координат относительно глобальной (отсчитывается от оси Y против часовой стрелки).

Код функции *setCoordinates()* приведён на рисунке 5.

```
void MyRobot::setCoordinates(Point2f& point, float localX, float localY)
{
    point.x = m_center.x + localX * cos(m_angle) + localY * sin(m_angle);
    point.y = m_center.y + localY * cos(m_angle) - localX * sin(m_angle);
}
```

Рисунок 5 — Код функции *setCoordinates()*

Проверка на пересечение роботом границ изображения выполняется во вспомогательной функции *checkBorders()*. В ней производится сравнение координат точки с размером изображения. Код функции приведён на рисунке 6.

```
void MyRobot::checkBorders(Point2f point, Mat image)
{
    if (point.x < 0 || point.x > image.cols || point.y < 0 || point.y > image.rows)
    {
        Speed stopSpeed;
        stopSpeed.x = 0;
        stopSpeed.y = 0;
        this->setSpeed(stopSpeed);
        this->setAngularSpeed(0);
        putText(image, "Border violation",
            Point2f(image.cols / 2, image.rows / 2), FONT_HERSHEY_PLAIN, 2, Scalar(0, 0, 255));
    }
}
```

Рисунок 6 — Код функции *checkBorders()*

Образующие линии робота выводятся на изображение с помощью функции OpenCV *line()*.

Вывод

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