Государственное образовательное учреждение высшего образования



*«Московский государственный технический университет*

*им. Н.Э. Баумана (национальный исследовательский университет)»*

*(МГТУ им. Н.Э. Баумана)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ФАКУЛЬТЕТ «СПЕЦИАЛЬНОЕ МАШИНОСТРОЕНИЕ»*

*КАФЕДРА «ПОДВОДНЫЕ АППАРАТЫ И РОБОТЫ»*

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА СТУДЕНТА НА ТЕМУ

«СРАВНЕНИЕ РАЗЛИЧНЫХ ТИПОВ ОПОРНЫХ МАРКЕРОВ»

|  |  |  |
| --- | --- | --- |
| Руководитель НИРС | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | (Макашов А.А.) |
|  | (подпись, дата) |  |
| Исполнитель НИРС,  студент группы СМ11-11М |  | (Андреев Е.В.) |

Москва, 2019 г.

РЕФЕРАТ

Отчёт на \_\_ стр., \_ ч., 20 рис., 11 источников, 3 таблицы.

СРАВНЕНИЕ РАЗЛИЧНЫХ ТИПОВ ОПОРНЫХ МАРКЕРОВ.

Перечень ключевых слов: опорный маркер,

Целью данной работы является

Результаты работы:

В данной работе были определены

Оглавление

[Введение 4](#_Toc27073474)

[1. Обзор существующих маркеров 6](#_Toc27073475)

[2. Описание методики сравнения 8](#_Toc27073476)

[3. Маркеры ARuCO 10](#_Toc27073477)

[1.1 Определение зависимости размера маркера от расстояния и разрешения камеры 10](#_Toc27073478)

[4. Маркер Pi-Tag 14](#_Toc27073479)

[5. Активный маркер 15](#_Toc27073480)

[2.1 Алгоритм детектирования светодиодов на основе ярокости 15](#_Toc27073481)

[2.2 Алгоритм детектирования светодиодов на основе преобразования Хаффа 17](#_Toc27073482)

[2.3 Алгоритм детектирования светодиодов на основе преобразования Хаффа 17](#_Toc27073483)

[3. Выбор камеры 18](#_Toc27073484)

[Заключение 18](#_Toc27073485)

[Список использованных источников 19](#_Toc27073486)

[Приложение 1. Исходный код примера алгоритма детектирования светодиодов 20](#_Toc27073487)

# Введение

В настоящее время всё большее применение находят АНПА (автономные необитаемые подводные аппараты). Одним из самых сложных вопросов при проектировании аппаратов подобного класса остаётся вопрос энергетики. Время автономной работы последних ограничено ёмкостями их энергетических элементов, необходимо обеспечить длительную автономность аппарата за счет использования достаточного количества энергоносителей, но при этом соблюсти ограничение по весу.

Актуальной задачей является продление времени автономной работы аппарата и увеличение периода нахождения его под водой. Для сокращения времени простоя при проведении подводных исследований, а также для тех работ, всплытие во время которых невозможно, предложен вариант стыковки АНПА с донной зарядной станции.

Наведение на донную зарядную станцию на больших расстояниях (свыше 10 метров) осуществляется по гидроакустическому маячку. Навигацию в непосредственной близи логично осуществлять с помощью видеокамер (ВК), установленных на АНПА, опорных маркеров определённого типа и алгоритмов технического зрения.

В данной работе объектом исследования являются опорные маркеры, использующиеся в робототехнике для навигации в пространстве. Предполагается, что аппарат оснащён всеми необходимыми датчиками, вопросы маневрирования в данной работе не рассматриваются.

Цель работы – изучение различных типов опорных маркеров и определение их применимости в подводной робототехнике.

Задачи:

- разработка методики сревнения различных типов опорных маркеров;

- исследование имеющихся готовых программных решений для обнаружения и определения маркеров;

- выбор сенсора и объектива ВК исходя из необходимости детектирования конкретного маркера;

- компоновка конструкции зарядной станции и выбранных средств наведения.

# 1. Обзор существующих маркеров

Опорные маркеры имеют уникальный дизайн и различные геометрические формы (рис. Рисунок 1) , однако все они могут быть автоматически обнаружены с помощью камеры. Системы определения маркеров используются в таких областях, как приложения дополненной реальности, медицина, космос и задачи, связанные с роботизацией. Разновидность области применения определяет критерии, которые характеризуют качественные свойства маркера и включают в себя такие критерии оценки, как устойчивость к перекрытию тенью, максимальное расстояние до маркера, ложноположительные и ложноотрицательные срабатывания, чувствительность детектирования к освещению и др.

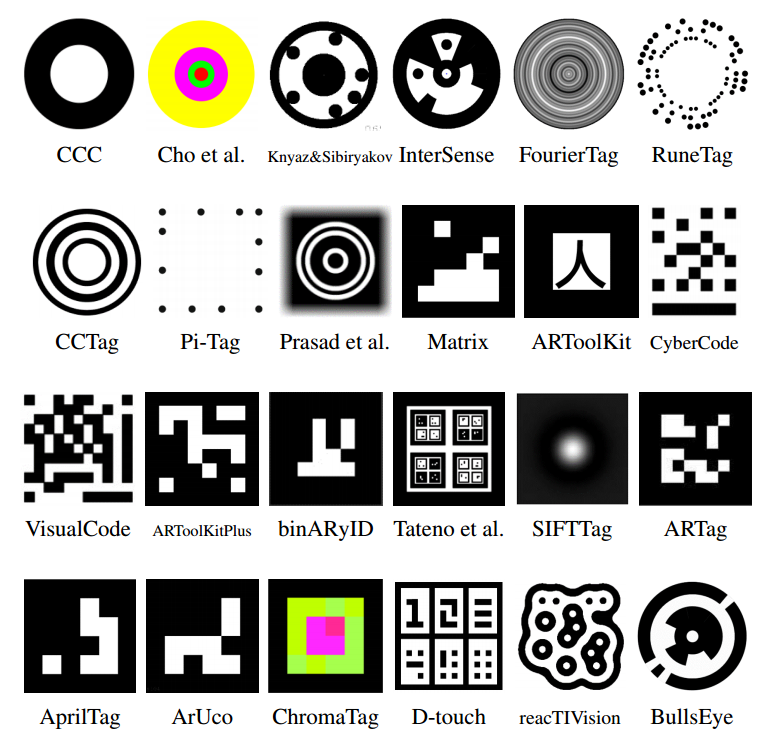


Рисунок 1 – Различные типы опорных маркеров

Опорным маркером может являться любая фигура. Однако на практике разработчики программного обеспечения мобильных роботов ограничины такими факторами, как разрешение видеокамеры, особенности цветопередачи и освещённости конкретной среды эксплуатации, вычислительная мощность оборудования, а также наличие или отсутствие открытых исходных кодов, содержащих алгоритмы детектирования. Как правило, обычно черно-белый маркер простой формы. Чаще это прямоугольник или квадрат со вписанным во внутрь идентификатором-образом.

В работе [1] проведено сравнение популярных маркеров ARTag, AprilTag и CALTag в отношении устойчивости детектирование при наличии вращения.

В статье [2] авторы описывают собственный опорный маркер и проводят его сравнение с имеющимися. Также описана общая последовательность операций над изображением при детектировании маркера. В работах [3,4] затрагиваются вопросы детектирования круглых маркеров, как цветных, так и чёрно-белых. Также рассматриваются алгоритмы оценки эллипсов и окружностей.

# 2. Описание методики сравнения

Прежде чем приступить к сравнению маркеров, необходимо составить методику оценки применимости конкретного типа в подводной робототехнике.

Для каждого типа определим 4 размера, условно обозначаемые в дальнейшем как «xs», «s» «m» и «l». Действительные габариты приведены в соответствующих таблицах ниже.

ВК настроена на максимально возможное разешение – 1080p, освещение – комнатное, оставалось неизменным на протяжении всех экспериментов.

Расположим маркеры на различных расстояниях от камеры и для каждого из них выясним возможность определения образца каждого размера. На рисунке Рисунок 2 показана схема эксперимента. Соответствующие расстояния указаны ниже:

*l1* = 5150 мм,

*l2* = 3520 мм,

*l3* = 1870 мм.

Также для каждого типа определим пороговые значения, определяющие границу идентифицируемости. Проделаем это для каждого идентифицируемого образца на расстояниях от *l1* до *l3*.

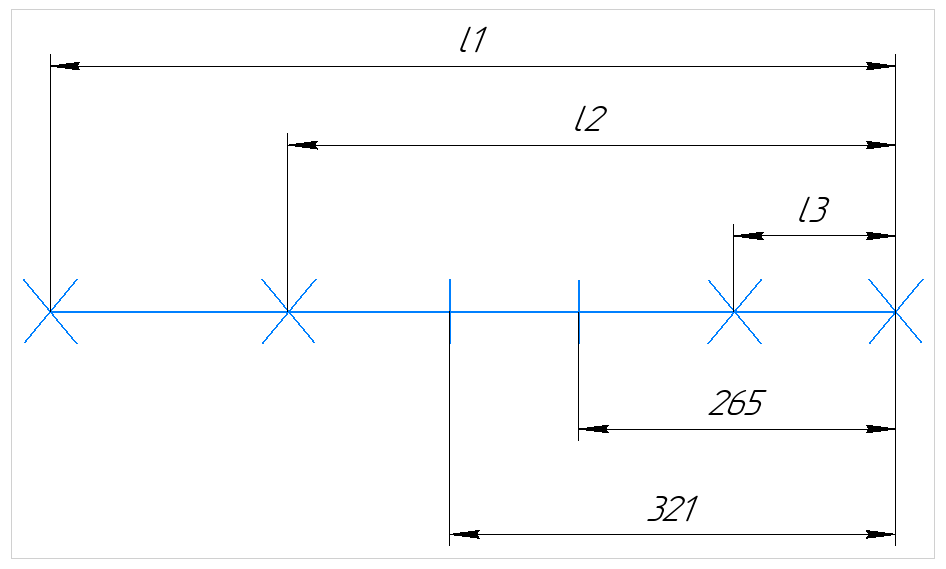


Рисунок 2 – Схема эксперимента

# 3. Маркеры ARuCO

Одним из наиболее популярных типов является ARuCo маркер, поскольку его реализация включена в стандартный пакет поставки билиотеки компьютерного зрения OpenCV (до версии 4) и вынесена в отдельный модуль пакета OpenCV-contrib начиная с версии 4.

## 1.1 Определение зависимости размера маркера от расстояния и разрешения камеры

Выявим зависимость габаритов маркера от дальности обнаружения и разрешения камеры экспериментальным путём (рис. Рисунок 3).

|  |  |  |
| --- | --- | --- |
|  |  |  |

Рисунок 3 – Проведение эксперимента

Требовалось определить возможность идентификации маркеров «xs» и «s» на выбранных расстояниях. Результаты эксперимента приведены в таблице Таблица 1.

Таблица 1.

Экспериментальные данные обнаружения

|  |  |  |
| --- | --- | --- |
| Размер | Расстояние | Определяется? |
| xs (60 x 60 мм) | *l1* | Нет |
| *l2* | Нет |
| *l3* | Да |
| s (90 x 90 мм) | *l1* | Нет |
| *l2* | Нет |
| *l3* | Да |
| m (130 x 130 мм) | *l1* | Да |
| *l2* | Да |
| *l3* | Да |

Пороговое расстояние (см. рис. Рисунок 2Рисунок 2 – Схема эксперимента) для маркера xs равно 2650 мм, при этом сам маркер имеет размеры 33 х 33 пикселя (рис. Рисунок 4).

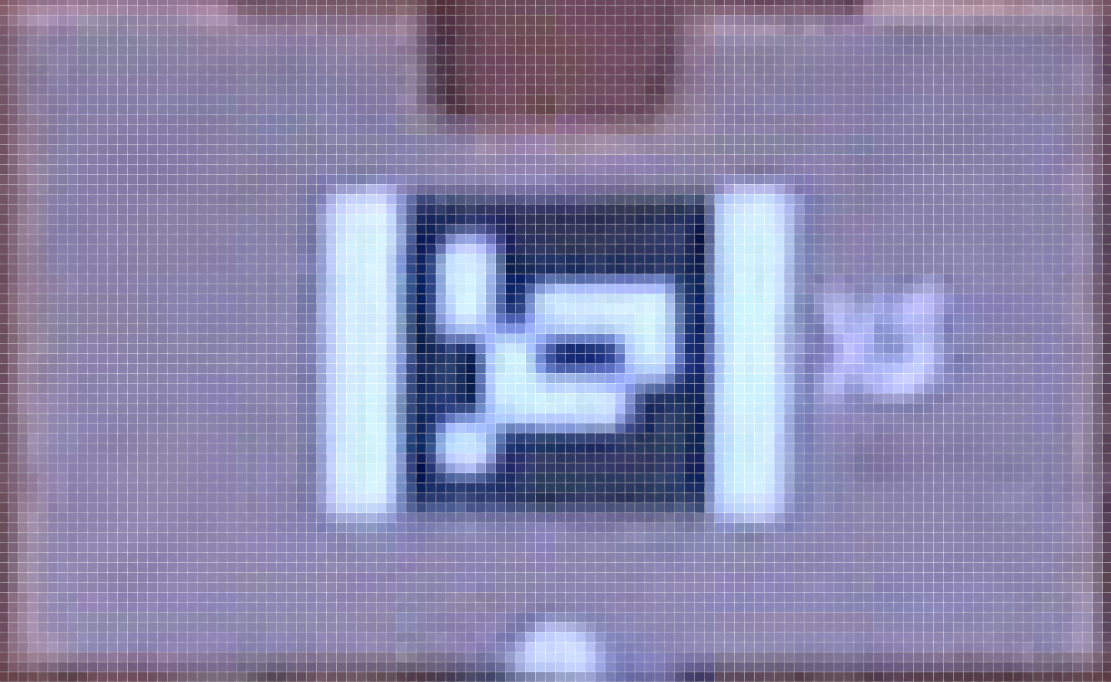


Рисунок 4 – Пикселизация маркера «xs»

Пороговое расстояние для маркера s равно 3210 мм, при этом сам маркер имеет размеры 40 х 40 пикселя (рис. Рисунок 5).

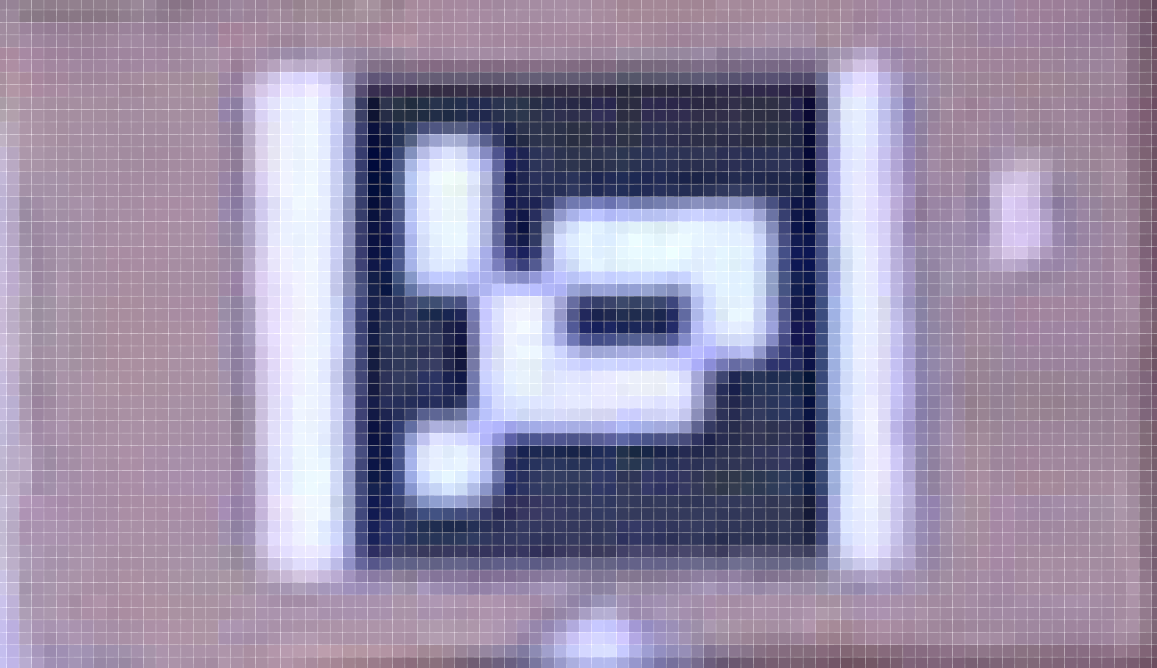


Рисунок 5 – Пикселизация маркера «s»

Выведем формулу, связывающую расстояние от камеры до объекта распознавания с габаритами последнего при заданном разрешении ВК. Очевидно, что размер макрека зависит прямо пропорционально от дальности и обратно пропорционально от разрешения. Влияние угла обзора камеры и его уменьшение при вереходе лучей из воздушной среды в водную, искажения, вызванные движением воды, а также возможность обнаружения маркера под углом на данном этапе не учитываем.

,

M – размер маркера, мм

k – масштабный коэффициент, пиксели

l - расстояние до маркера

r – разрешение камеры по ширине кадра

, отсюда .

Примем.

Таким образом, в идеальных условиях габариты опорного маркера, необходимые для его обнаружения с 10 метров:

,

что является неудовлетворительным результатом.

# 4. Маркер Pi-Tag

Опорный маркер Pi Tag (см. рис. Рисунок 6) представляет собой набор чёрных точек одинакового размера, расположенных на белом поле. Алгоритм детектирования представлен в виде открытых исходных кодов [] и является адаптированной версией пакета «cob fiducials» [] по определению опорных маркеров, входящего в состав ROS [].

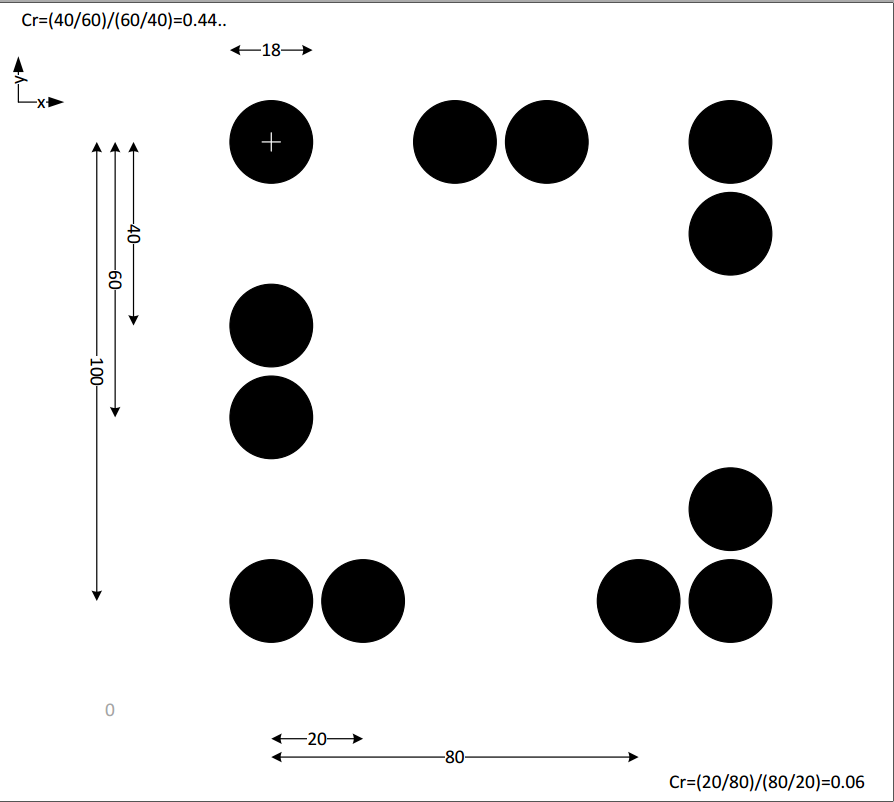


Рисунок – Пример маркера Pi Tag

Результаты эксперимента приведены в таблице Таблица 2.

Таблица 2

Экспериментальные данные обнаружения

|  |  |  |
| --- | --- | --- |
| Размер | Расстояние | Определяется? |
| xs (55 x 55 мм) | *l1* | Нет |
| *l2* | Нет |
| *l3* | Нет |
| s (85 x 85 мм) | *l1* | Нет |
| *l2* | Нет |
| *l3* | Нет |
| m (120 x 120 мм) | *l1* | Нет |
| *l2* | Нет |
| *l3* | Да |

Плихие результаты работы эксперимента заключаются во внутреннем устройстве алгоритма детектирования. Понимание последовательности операций даёт рисунок .

|  |  |
| --- | --- |
| C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\00.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\20.png |
| C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\30.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\40.png |
| C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\49.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\50.png |
| C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\60.png | |

Разработчики данной библиотеки не использовали какие-либо алгоритмы кластеризации для ещё большего отсеивания эллипсов. Помимо прочего данный тип маркеров нуждается в тонкой настройке параметров, отвечающих за наибольший и наименьший размеры эллипса, а также за соотношение сторон фигуры. Для того чтобы охватить все возможные варианты детектирования, включая обнаружение на большем расстоянии, необходимо заложить параметры, обеспечивающие наибольший диапазон вариантов. Однако в этом случае производительность обнаружения оставляет желать лучшего – от нескольких секунд до 3-х минут (!), поскольку программа вынуждена перебирать количество вариантов, равное числу сочетаний Сn4, где n – все обнаруженные эллипсы.

Дальнейшее исследование этого маркера считаем нецелесообразным.

# 5. Активный маркер

Хорошим опорным объектом может являться светоизлучающий диод (СИД), поскольку он имеем малые размеры и будет в гораздо меньшей степени подвержен влиянию различных донных течений. К тому же, СИД очень близок к точечному источнику света, что облегчает детектирование.

## 2.1 Алгоритм детектирования светодиодов на основе ярокости

Предлагается использовать следующую последовательность действий для обнаружения и определения положения СИД:

1. Получение изображения в оттенках серого.
2. Извлечение порогового изображения.
3. Фильтрация и сглаживание порогового изображения.
4. Нахождение контуров объектов с помощью детектора Канни [].
5. Определене описывающего прямоугольника для каждого контура.
6. Нахождение эллипса для каждого контура.
7. Вычисление центра как полусуммы координат фокусов эллипса.

|  |  |  |
| --- | --- | --- |
| C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\s1.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\s2.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\s3.png |
| C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\s4.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\s5.png | C:\Users\Eugene\AppData\Local\Microsoft\Windows\INetCache\Content.Word\s6.png |

Рисунок 7 – Последовательность преобразования изображения для обнаружения положений светодиодов

Пороговое изображение определяется по уровню яркости, который задан в виде постоянного значения. В качестве сглаживающей операции используется морфологическое преобразование «opening» [], которое является комбинацией преобразований «erode» [] и «dilite» []. Матрица преобразования прямоугольная размера 5х5.

Детектор Канни призван обнаруживать границы объектов на изображении. Для его работы необходимо задать наибольшее и наименьшее ожидаемые значения яркости пикселей на границах обкъектов (в нашем случае контуров).

Исходный код примера исползования алгоритма приведён в приложении 1.

## 2.2 Алгоритм детектирования светодиодов на основе преобразования Хаффа

## 2.3 Самодельный активный маркер

Самодельный активный маркер представляет собой основу, на которой закреплены 4 СИД. В основе были проделаны отверстия под маркеры разных размеров (рисунок ).

Если пассивные маркеры «страдают» от эффекта размытия, то светодииодные маркеры сами вызывают подобное явление.

Результаты эксперимента приведены в таблице Таблица 3.

Таблица

Экспериментальные данные обнаружения

|  |  |  |
| --- | --- | --- |
| Размер | Расстояние | Определяется? |
| xs (60 x 60 мм) | *l1* | Нет/ |
| *l2* | Нет/ |
| *l3* | Нет/ |
| s (90 x 90 мм) | *l1* | Нет/ |
| *l2* | Нет/ |
| *l3* | Да |
| m (130 x 130 мм) | *l1* | Нет/ |
| *l2* | Нет/ |
| *l3* | Нет/ |

# 3. Выбор камеры

# Заключение

# Список использованных источников

# Приложение 1. Исходный код примера алгоритма детектирования светодиодов

//https://docs.opencv.org/4.1.1/de/d62/tutorial\_bounding\_rotated\_ellipses.html

//https://docs.opencv.org/4.1.1/d3/dbe/tutorial\_opening\_closing\_hats.html

#include <opencv2/opencv.hpp>

#include <opencv2/aruco.hpp>

using namespace cv;

using namespace std;

int threshold\_value = 245;

//int threshold\_type = 3;

int const max\_value = 255;

int const max\_type = 4;

int const max\_binary\_value = 255;

float treshold\_area = 30 \* 20;

string path = "../../sources/";

string dir = "";

string file = "led\_blue.jpg";

string full\_path = path + dir + file;

Mat image, image\_gray, hough, tresh, tresh\_opening, canny\_output, drawing;

RNG rng(12345);

vector<Vec3f> circles;

void morphology\_Operation(int morph\_operator, Mat& src, Mat & dst)

{

int morph\_elem = MORPH\_RECT;

int morph\_size = 5;

Mat element = getStructuringElement(morph\_elem, Size(2 \* morph\_size + 1, 2 \* morph\_size + 1), Point(morph\_size, morph\_size));

morphologyEx(src, dst, morph\_operator, element);

imshow("Morph operation", dst);

}

int main(int argc, char\*\* argv)

{

image = imread(full\_path, IMREAD\_COLOR);

cvtColor(image, image\_gray, COLOR\_BGR2GRAY); // Convert the image to Gray

//GaussianBlur(image, image\_copy, Size(13, 13), 0, 0);

image.copyTo(hough);

HoughCircles(image\_gray, circles, HOUGH\_GRADIENT, 1,

image\_gray.rows / 32, // change this value to detect circles with different distances to each other

100, 30, 20, 50 // change the last two parameters

// (min\_radius & max\_radius) to detect larger circles

);

for (size\_t i = 0; i < circles.size(); i++)

{

Vec3i c = circles[i];

Point center = Point(c[0], c[1]);

// circle center

circle(hough, center, 1, Scalar(0, 100, 100), 3, LINE\_AA);

// circle outline

int radius = c[2];

circle(hough, center, radius, Scalar(255, 0, 255), 3, LINE\_AA);

}

threshold(image\_gray, tresh, threshold\_value, max\_binary\_value, THRESH\_TOZERO);

morphology\_Operation(MORPH\_OPEN, tresh ,tresh\_opening);

//Canny(image\_gray, canny\_output, threshold\_value-30, 255);

//Canny(tresh, canny\_output, 0, 255);

Canny(tresh\_opening, canny\_output, 0, 255);

vector<vector<Point> > contours;

findContours(canny\_output, contours, RETR\_TREE, CHAIN\_APPROX\_SIMPLE, Point(0, 0));

vector<RotatedRect> minRect(contours.size());

vector<RotatedRect> realRect;

vector<RotatedRect> minEllipse;

for (size\_t i = 0; i < contours.size(); i++)

{

minRect[i] = minAreaRect(contours[i]);

if (contours[i].size() > 5 && (float(minRect[i].size.width) \* float(minRect[i].size.height) > treshold\_area)) {

realRect.push\_back(minRect[i]);

minEllipse.push\_back(fitEllipse(contours[i]));

}

}

drawing = Mat::zeros(canny\_output.size(), CV\_8UC3);

for (size\_t i = 0; i < realRect.size(); i++)

{

Scalar color = Scalar(rng.uniform(0, 256), rng.uniform(0, 256), rng.uniform(0, 256));

// contour

//drawContours(drawing, contours, (int)i, color);

// ellipse

ellipse(drawing, minEllipse[i], color, 2);

// rotated rectangle

Point2f rect\_points[4];

realRect[i].points(rect\_points);

for (int j = 0; j < 4; j++)

{

line(drawing, rect\_points[j], rect\_points[(j + 1) % 4], color);

}

}

imshow("Original", image);

imshow("Grayscale", image\_gray);

imshow("Treshold", tresh);

imshow("Hough Detector", hough);

imshow("Canny output", canny\_output);

imshow("Contours", drawing);

waitKey(0);

return 0;

}