электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

 # 06, июнь 2016

УДК 681.531.2

Распознавание объектов с помощью телекамеры

Воронин А.В., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Специальная робототехника и мехатроника»

Научный руководитель: Рубцов В.И., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Специальная робототехника и мехатроника»
kafsm7@sm.bmstu.ru

Введение

Системы технического зрения имеют широкое применение в робототехнике. Движение многих мобильных роботов осуществляется системой управления движением, которая анализирует изображения с видеокамеры.

Рис.1. E-RevoBrushless (#56087)

Требования к алгоритму:

1. Параметры входного сигнала:

1.1. размер кадра: не менее 340x270 пикселей;

1.2. частота кадров: не менее 25 кадров/сек.

2. Освещенность:

2.1. min освещенность сцены: не менее 100 люкс;

2.2. max освещенность сцены: не более 50000 люкс;

3. Наличие перспективы на фотографии;

4.*Время распознавания препятствия: не более 7 сек;*

Постановка задачи

Главной задачей является разработка алгоритма распознавания объектов для автономного движения мобильного колесного робота по пересеченной местности на базе шасси E-RevoBrushless (#56087), со временем распознавания кадра не более 7 сек., и разрешением телекамеры не менее 340x270 пикселей.

Автомодель выполнена в масштабе 1/8(рис. 1) – обладает достаточными габаритами и дорожным просветом для перемещения по пересеченной местности.

В данной работе объектом распознавания являлся куб.

Этапы обработки алгоритма

Для разработки программного обеспечения была выбрана программа математического моделирования MATLAB, поскольку обладает большим количеством математических пакетов, позволяющие решать большое количество разнообразных задач, возникающих в различных областях человеческой деятельности.

Обработка изображения проходит в несколько этапов. Это находит отображение в разделении программы на отдельные функции, каждая из которых получает в качестве входных данных результат работы предыдущих функций и производит собственную обработку.

Алгоритм состоит из следующих этапов:

1. предобработка изображения:
   - фильтрация;
2. выделение контуров на изображении;
3. сегментация линий на изображении;
4. алгоритм распознавания объектов.

Предобработка изображения

Фильтрация

В изображениях, получаемых с видео датчика могут наводиться помехи, вызванные зернистостью самой камеры, различного рода искажения и смазы, засвеченность или

наоборот слишком затемненное изображение, поэтому приходиться проводить фильтрацию изображения с целью повышения качества входного изображения, и, следовательно, повышения вероятности обнаружения контуров объекта. Существуют различные методы улучшения качества входного изображения. В данном алгоритме реализованы фильтры, расположенные в такой последовательности:

- стабилизация уровня яркости
- фильтрация высоких и низких частот
- повышение контрастности
- Винеровская фильтрация

Поскольку съемка может проходить и в пасмурный день, и в яркий солнечный день, то входные изображения могут быть затемненными или, соответственно, пересвеченными, из-за чего необходимо провести ***выравнивание уровня яркости на изображении***. Для этого исходное изображение представляется в формате данных *double*(числа с плавающей запятой) в диапазоне [0,1], в котором 0,5 – рекомендуемый уровень яркости, и соответствует 100%. Вычисляется среднее значение яркости всех пикселов, входящих в изображение, и отклонение получившегося значения в процентах от рекомендуемого. На полученную величину происходит *гамма-коррекция* изображения.

Фильтрация высоких и низких частот осуществляется с помощью последовательного выполнения фильтров Гаусса и Лапласа, так называемого Лапласиана - Гауссиана. Фильтр Лапласа (фильтр высоких частот) повышает резкость, а Гаусса (фильтр низких частот) ограничивает появление дополнительной зернистости на изображении. [7]

Винеровская фильтрация используется для подавления аддитивного гауссова белого шума. Данный алгоритм основан на статистических оценках фрагментов изображения в пределах скользящего окна размера C × E пикселей.

Для всех положений скользящего окна с центральным пикселем и координатами $n_1 \times n_2$ вычисляются:

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)^2 - \mu^2$$

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}\ (a(n_1, n_2) - \mu),$$

где F −среднее значение яркости, σ^2 – дисперсия, $a(n_1, n_2)$ – значение яркости вычисляемого пикселя в скользящем окне, $b(n_1, n_2)$ − значение яркости центрального пикселя. Данная формула применяется не рекурсивно для всех положений скользящего окна.

Если мощность v гауссова белого шума не задана, то она оценивается как среднее из всех σ в пределах скользящего окна.[2]

Выделение контуров на изображении

После применения всех фильтров на этапе предобработки, на изображении выполняется сегментация линий объекта.

Для этого можно провести **бинаризацию изображения** по заданному порогу. Пикселям, на изображении которые имеют значение яркости больше пороговой, будет присвоено значение 1 в формате double или 255 в uint8. Остальным пикселям будет присвоено значение 0. Таким образом, происходит бинаризация изображения, на выходе которой получается чёрно-белое изображение без градаций серого. Данный метод прост в реализации, не требует вычислительных ресурсов, обладает высокой скоростью выполнения.

К недостаткам бинаризации можно отнести появление ошибок в различных ситуациях, например: блики на дорогах, яркие объекты на снимке и т.д.

В данном алгоритме был использован **метод Кенни**, который осуществляет поиск границ объектов − участки изображения, в которых есть перепад яркости. Этот метод определяет контуры путём поиска локальных максимумов градиента яркости пикселов. Перепады яркости (градиент) ищутся с помощью фильтрации по каждой из осей одномерным фильтром Лапласиан–Гауссиана.

При вычислении градиента мы сталкиваемся с необходимостью выбора порогового значения градиента, который определит, какие из контуров нам отсеять, а какие принять, поскольку далеко не все границы представляют нужную нам информацию. При этом мы сталкиваемся с риском потери части, возможно, важной нам информации. Поэтому мы принимаем два значения порогового градиента: одно − для слабо выраженных границ, другое – для ярко выраженных. Причём слабые контуры принимаются расчёт только

тогда, когда они соединяются с сильными, т.е. образуют непрерывную кривую.

Достоинством данного метода является возможность определения слабых границ.[1]

Сегментация линий на изображении

На предыдущем этапе были получены точки, находящиеся на контурах объектов, теперь надо найти линии на изображении (поскольку объектом сегментации являются линии квадрата) и уже потом отсортировать их по определенным признакам. Для этого применяется **преобразование Хафа**[5]. Алгоритм преобразования Хафа использует массив, называемый *аккумулятором (H)*, для определения присутствия прямой y = mx + b на изображении BW. Данный массив возвращает функция [H, theta, rho] = hough(BW), где параметры theta (в градусах) и rho представляют массивы значений, на основе которых генерируется матрица преобразований Хафа. Размерность аккумулятора равна количеству неизвестных параметров пространства Хафа. Два измерения аккумулятора соответствуют квантизированным значениям параметров m и b. Для каждой точки и её соседей алгоритм определяет, достаточен ли вес границы в этой точке. Если да, то алгоритм вычисляет параметры прямой (для каждого theta по формуле r = xcosϴ + ysinϴ вычисляется rho) и увеличивает значение в ячейке аккумулятора, соответствующей данным параметрам. Причем значением каждой ячейки является число точек плоскости X-Y проходящих через

прямую с координатами (rho и theta) данной ячейки. Потом, найдя ячейки аккумулятора с максимальными значениями функцией houghpeaks в пространстве аккумулятора, могут быть определены наиболее подходящие прямые. Так как полученные прямые не содержат информацию о длине, следующим шагом является нахождение частей изображения, соответствующих найденным прямым. С помощью преобразований из параметров rho и theta вычисляются начальные и конечные координаты линий. Из-за ошибок на этапе определения границ фигур в пространстве аккумулятора также будут содержаться ошибки. Это делает поиск подходящих линий нетривиальным.[5]

Алгоритм распознавания квадрата

Выбранная телекамера работает с 2-D пространством, что позволяет решать задачу распознавания не куба, а квадрата.

После сегментации на изображении могут регистрироваться линии, не являющиеся контурами квадрата, поэтому линии надо отфильтровать по ряду признаков. Но прежде чем фильтровать полученный массив линий, определяется уровень качества снимка по количеству найденных линий. По умолчанию программа работает в режиме для

изображений с плохим качеством, но, если на снимке присутствуют пики, полученные преобразованием Хафа, то программа переходит в режим для *изображений хорошим качеством*. При этом программа возвращается на этап выделения контуров, и повторно выделяются границы методом Кенни, но уже с более высоким порогом thresh. Далее на бинарном изображении границы объектов увеличиваются винеровской фильтрацией для лучшего обнаружения линий методом Хафа, пороги которого также подняты. Дополнительно изображение проходит через адаптивный медианный фильтр для подавления помех, полученных в результате усиления контуров.

Для селектирования квадрата используются только линии красного цвета. Из этих линий выбираются только те, которые удовлетворяют условиям перечисленные ниже:

1. Проверяемая линия должна иметь параллельную и перпендикулярную себе линию.
2. Линии должны быть одинаковой длины.
3. У перпендикулярных линий должны совпадать начальные координаты.
4. Если все предыдущие условия выполнены, то текущая линия считается стороной квадрата, так как у квадрата 4 стороны, то это условие должно, выполнятся 4 раза.

Описание программы

Написана программа в среде MATLAB по распознаванию изображений.

В качестве объекта выступает красный куб. Алгоритм работы программы представлен на рис.2.

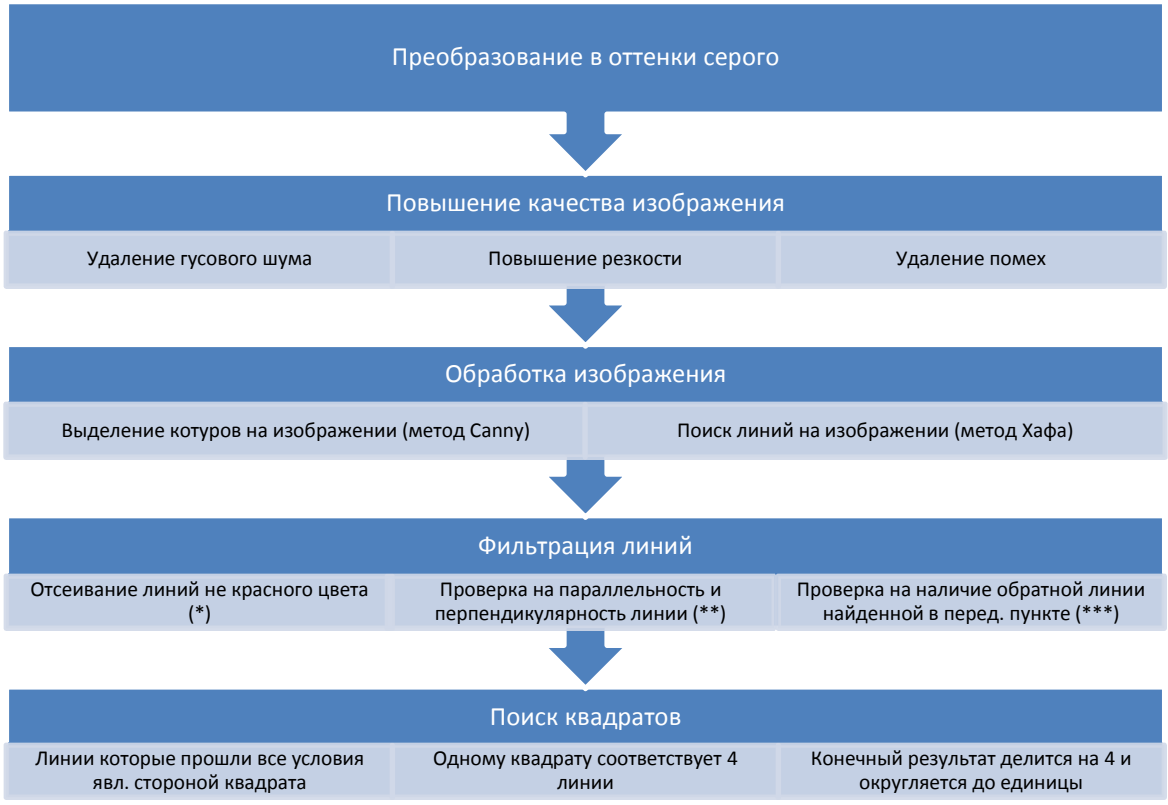

Рис.2. Алгоритм работы программы

В результате выполнения программы на экран компьютера выводятся окна с выделенными линиями, и найденными квадратами на изображениях.

Интерфейс в среде MATLAB

В ходе выполнения программы на экран выводятся окна с основными этапами:

1. Исходное изображение (рис.3).

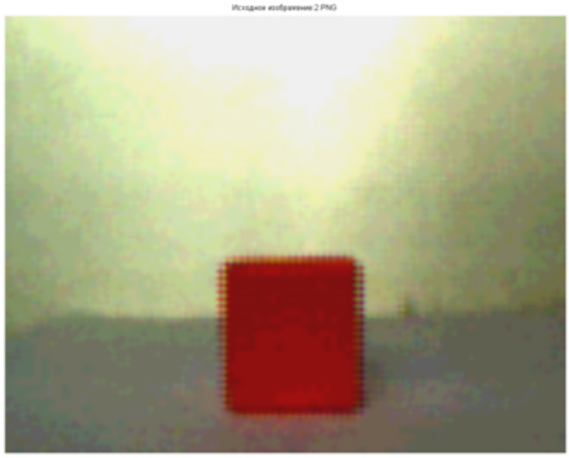

Рис. 3. Исходное изображение

2. Изображение, полученное в результате обработки (рис.4).

Рис.4. Результат фильтрации

3. Изображение, обработанное методом Кенни (рис.5).

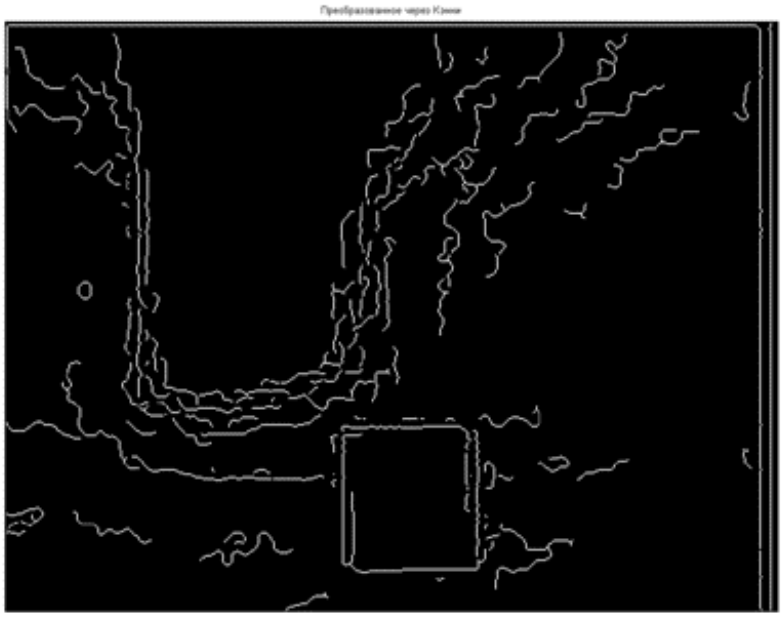

Рис. 5. Обработка изображения методом Кенни

4. Окно с выводом пространства Хафа (рис.6).

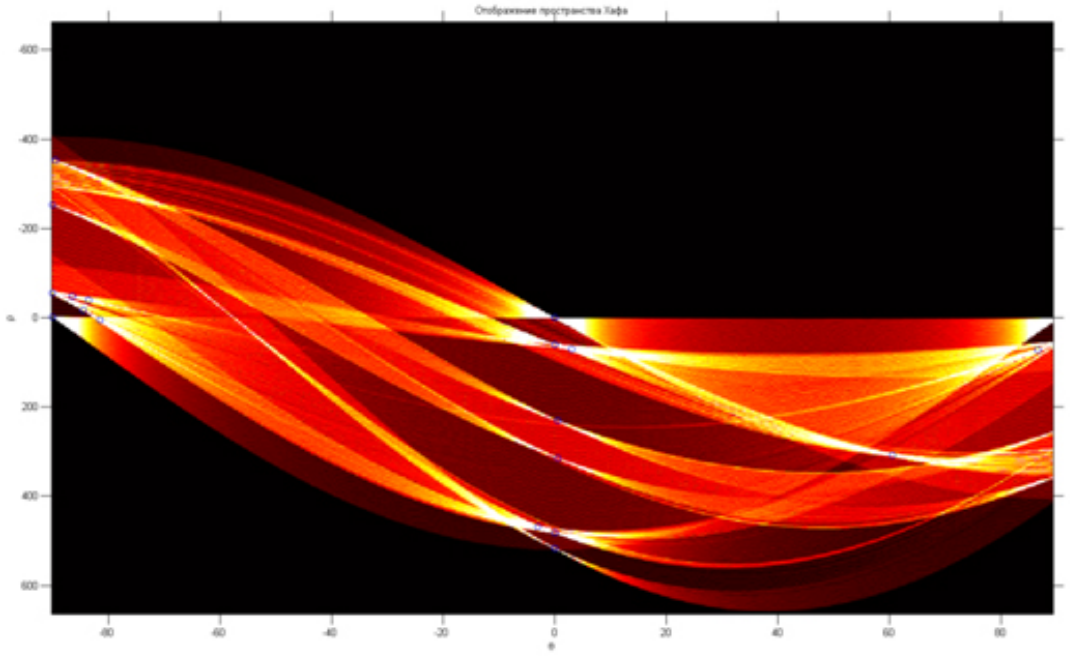

Рис. 6. Вывод пространства Хафа

5. Исходное изображение с выделенными линиями. Подсвеченные черным цветом линии, интерпретируемые как красные, зеленым - обозначаются линии не красного цвета, красным – линии которые непосредственно участвовали в обнаружении квадрата (рис.7).

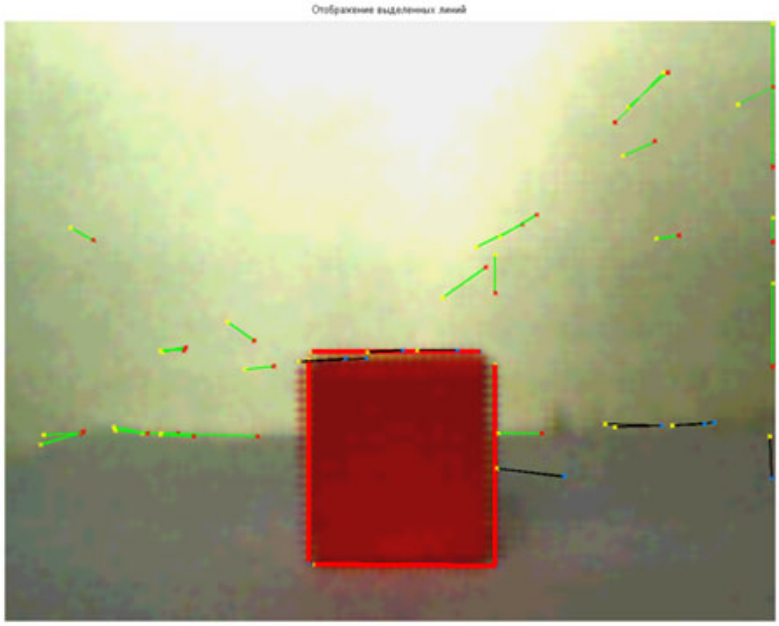

Рис. 7. Отображение выделенных линий

6. Окно с количеством найденных квадратов (рис.8). Полученный результат можно сохранить в формате “tiff”. Он сохраняется как [saved_завание файла] в папке с программой.

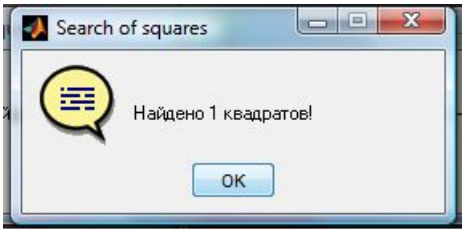

Рис. 8. Отображение числа найденных квадратов

Тестирование программы

Проведение тестов:

При движении мобильного робота могут изменяться условия внешней среды (освещения, дорожного покрытия). Следовательно, программа должна функционировать в широком диапазоне внешних условий. Программа должна работать при разных **условиях освещения**, при движении робота по реперам, могут накапливаться некоторые ошибки связанные с ориентацией в пространстве, поэтому **угол поворота** робота по отношению к объекту может меняться изменяться от 0 до 60 градусов .Также объект может быть **частично загорожен** (от 0% до 50%). Поверхность, по которой движется робот, может быть не ровной, поэтому **объект может быть наклонен относительно робота**. При передвижении робота объект может быть удален от робота или наоборот приближен, вследствие этого объект нужно распознавать с **разных расстояний**. Влияние всех пяти факторов необходимо проверить. Для проверки работы программы, было проведено математическое моделирование.

Для оценки работоспособности программы, тестируемые изображения были получены с камеры NXTCAM-v3. Полученные изображения делились на две группы: с наличием красного куба на фотографии для определения погрешности первого рода и изображения с отсутствием красного куба или другого объекта на фотографии для определения погрешности второго рода. Все тесты кроме теста на освещенность производились в дневное время. Для отработки отдельных элементов фон был загорожен белыми листами. Тесты проходили на разноцветных кубиках со сторонами 8см, и на других различных фигурах (красный цилиндр, параллелепипедах разного цвета, фигура в виде елки зеленого цвета и т.д.).

Тестирование проводилось на следующей конфигурации:

Операционная система: Windows 7

Среда: MATLAB 7.10.0.499 (R2014a)

Формат исходного изображения: tiff, jpeg, png

Разрешение изображения: около 495X388

Цветность изображения: цветное, черно-белое

Освещение:

При проведении тестов параметры освещения были следующими:

1. *Дневное освещение (40%-60%)*
2. *Комнатное освещение (30%-40%)*
3. *Лампа (15%-30%)*
4. *Фонарь (9%-15%)*

Кроме того освещение замерялось специальным датчиком яркости. В описании освещения введены значения, снимаемые с данного датчика в процентах.

Угол поворота объекта:

При проведении тестов по изменению угла поворота тестируемый объект, был расположен на некотором расстоянии от робота, и в начальном положении перпендикулярен ему. Куб поворачивали относительно камеры от 0° до 60° градусов с шагом в 5 ° (см. знач. в таблице 2).

Не полная поверхность объекта:

При проведении тестов поверхность куба закрывали от 0% до 50% его площади с шагом в 5% (см. знач. в таблице 2).

Расстояние между роботом и объектом:

При проведении тестов по расстоянию куб перемещали на расстояния от 35см до 80см от камеры (см. знач. в таблице 2).

Неровная поверхность:

При проведении тестов по неровной поверхности изменяется угол между плоскостью, на которой лежит куб и им самим (рис. 9). Значения угла изменялись от 3° до 30° с шагом в 3° (см. знач. в таблице 2).

Рис. 9 Неровная поверхность(схематично)

Выводы, сформированные на основе проведенных тестов

По Освещению:

При работе с данной программой и использовании камеры NXTCAM-v3, следует не допускать бликов. Если освещение около 9%, то свет должен быть не точечным, а размыт. При использовании искусственного света увеличивается вероятность ошибки. С уменьшением освещенности растет количество помех, следовательно, увеличивается вероятность ошибки в определении цвета, и видимости контура квадрата. Когда объект находиться на контрастном фоне, квадрат также плохо опознаваем. Таким образом, если показания с датчика освещение 9%-15%,то следует, включите фонарь.

Расстояние между роботом и объектом:

Не рекомендуется использовать программу для работы на расстояниях более 50см при плохом освещении т.к. изображения, получаемые с камеры плохого качества. Минимальное расстояние от квадрата до камеры 25см.

По углу поворота:

При повороте куба относительно камеры более 50° увеличивается вероятность не распознавания объекта. Не рекомендуется использовать при плохом освещении.

По неполной поверхности объекта:

При закрытии поверхности более 20% значительно увеличивается вероятность ошибки. Более 35% квадрат вообще не распознается. Зависит от освещенности.

По неровной поверхности:

Если угол между плоскостью и кубом более 10° вероятность ошибки растет с увеличением угла. Зависит от освещенности.

Рекомендации по использованию программного обеспечения

Общие рекомендации:

При дневном освещении программа работает с наименьшей вероятностью ошибок.

По освещению:

Программа устойчиво работает при дневном освещении, ровном контрастном фоне (50%-60% освещенности по датчику яркости).

По расстоянию между роботом и объектом:

Рабочий режим от 30см до 80см от камеры при нормальном дневном свете.

По углу поворота объекта:

Наиболее подходящий режим для работы программы 0°-45° при нормальном дневном свете.

По неполной поверхности объекта:

Если закрыто менее 15% площади поверхности куба при нормальном дневном свете.

По неровной поверхности:

Наиболее благоприятный режим для работы программы, если угол между плоскостью и кубом 0°-10° при нормальном дневном свете.

Общие результаты:

Результаты, полученные экспериментальным путем, приведены в таблицах 1-2.

Таким образом, в типовых режимах работы робота (освещение 40%-60%, угол поворота объекта 0°-45°, расстояние между роботом и объектом 30см-80см, закрытая поверхность объекта менее 15%, не ровная поверхность 0°-10°), математическое моделирование показало, что ошибки первого второго рода не превышают 10%. Данная программа может работать с более высоким кпд (увеличение расстояние на котором можно распознать репер, увеличение угла между плоскостью и репером), если увеличить качество входных изображений, и изменить параметры программы.

Выполнена серия экспериментов разработанного алгоритма и занесена в таблицы.

Таблица 1

Экспериментальная таблица 1

Параметры	1-го рода	2-го рода
Освещение	70%	70%
Угол поворота объекта	90%	90%
Не полная поверхность объекта	60%	90%
Расстояние между роботом и объектом	100%	90%
Неровная поверхность	50%	80%
Результат	74%	84%

Таблица 2

Экспериментальная таблица 2

Фотографии					
для определения погрешности первого рода			для определения погрешности второго рода		
Описание	№	Оценка	Описание	№	Оценка
Освещение					
Комнатное освещение					
35%	1	-	34%	1	+
34%	2	-	33%	2	-
33%	3	+	33%	3	-
34%	4	+	33%	4	-
Лампа					
19%	5	+	18%	5	+
19%	6	+	18%	6	+
16%	7	+	18%	7	-
Фонарь					
10%	8	-	9%	8	+
10%	9	-	9%	9	+
9%	10	+	9%	10	+
Дневное освещение					
60%	11	+	47%	11	+
54%	12	+	47%	12	+
52%	13	+	39%	13	+
48%	14	+	52%	14	+
Угол поворота					
0°	1	+	0°	1	+
10°	2	+	10°	2	-
15°	3	+	15°	3	+
20°	4	+	20°	4	+
25°	5	+	25°	5	+
30°	6	+	30°	6	+
45°	7	+	45°	7	+
50°	8	+	50°	8	+
55°	9	-	55°	9	+
60°	10	+	60°	10	+
Не полная поверхность объекта					
0%	1	+	0%	1	-
10%	2	+	10%	2	+
15%	3	+	15%	3	+
20%	4	+	20%	4	+
25%	5	+	25%	5	+
30%	6	+	30%	6	+

35%	7	-	35%	7	+
40%	8	-	40%	8	+
45%	9	-	45%	9	+
50%	10	-	50%	10	+
Расстояние между роботом и объектом					
35см	1	+	35см	1	-
40см	2	+	40см	2	+
45см	3	+	45см	3	-
50см	4	+	50см	4	+
55см	5	+	55см	5	+
60см	6	+	60см	6	+
65см	7	+	65см	7	+
70см	8	+	70см	8	-
75см	9	+	75см	9	+
80см	10	+	80см	10	+
			35см	11	+
			40см	12	+
			45см	13	+
			50см	14	+
			55см	15	+
			60см	16	+
			65см	17	+
			70см	18	+
			75см	19	+
			80см	20	+
Неровная поверхность					
3°	1	+	3°	1	+
6°	2	+	6°	2	+
9°	3	+	9°	3	+
12°	4	-	12°	4	+
15°	5	+	15°	5	-
18°	6	-	18°	6	-
21°	7	-	21°	7	+
24°	8	-	24°	8	+
27°	9	+	27°	9	+
30°	10	-	30°	10	+

Заключение

Разработан алгоритм распознавания объектов. Проведено компьютерное моделирование разработанного алгоритма в среде математического моделирования Mathworks MATLAB. Выполнена серия экспериментов разработанного алгоритма.

Список литературы

[1]. MathWorksfspecial. Available at: http://www.mathworks.com/help/images/ref/fspecial.html(дата обращения 15.01.2016)

[2]. Math Works wiener2. Available at: http://www.mathworks.com/help/images/ref/wiener2.html(дата обращения 15.01.2016)

[3]. Контрастирование с гамма-коррекцией. Режим доступа: http://matlab.exponenta.ru/imageprocess/book3/10/imadjust.php(дата обращения 15.01.2016)

[4]. Википедия. Преобразование Хафа. Режим доступа: https://ru.wikipedia.org/wiki/Преобразование_Хафа(дата обращения 15.01.2016)

[5]. Википедия. Прямая. Режим доступа: https://ru.wikipedia.org/wiki/%CF%F0%FF%EC%E0%FF(дата обращения 15.01.2016)

[6]. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, автоматики // Современные технологии автоматизации. 2010. № 1. С. 615-620.

[7]. Машков К.Ю., Рубцов В.И., Рубцов И.В. Состав и характеристики мобильных роботов. М.: МГТУ им. Н.Э. Баумана, 2014. 76 с.