**Алгоритмы распознавания объектов для строительства инфраструктуры группой роботов в автономном режиме**

**Автор:** Кудряшов Игорь Сергеевич ,ГБОУ 1505 , г. Москва, 10 класс

**Научный Руководитель:** Ветюков Дмитрий Алексеевич

**Москва**

**2017 г.**

**Содержание**

Введение…………………………………………………………………………………………….2

Постановка задачи…………………………………………………………………………………4

Разработка алгоритмов…………………………………………………………………………….4

Заключение………………………………………………………………………………………….9

Список литературы……………………………………………………………………….…..……11

Приложение…………………………………………………………………………………………12

**Введение**

У человечества уже довольно давно появилась идея освоения дальнего космоса. Решение данной задачи требует строительства перевалочных пунктов для обновления и дозаправки летательных аппаратов. Одним из возможных месторасположений строительства перевалочного пункта является Луна. Очевидно, что для полноценного космодрома на поверхности Луны человечество еще не доросло, поэтому необходима постройка обитаемых лунных баз.

Основные аргументы в пользу строительства обитаемой лунной базы сводятся к следующему:

* Такая база удобна для ведения непрерывного глобального контроля всей поверхности Земли и окружающего ее космического пространства;
* С нее возможно проведение уникальных астрофизических экспериментов;
* Малая сила тяжести и умеренные затраты энергии для отлета с Луны в сочетании с ее близостью к Земле создают благоприятные возможности вовлечения лунных ресурсов в сферу космического производства, которое может быть организовано на геоцентрических и селеноцентрических орбитах. При этом важно отметить, что первичную обработку лунного сырья целесообразно производить на промышленных установках, расположенных на Луне.

1. ***Лунная колония программы «Apollo»***

Теперь уже мало кто вспоминает, что американская космическая программа «Аполлон» закончилась совсем не так, как задумывалась. В период с 1970 по 1972 год планировалось провести восемь пилотируемых экспедиций на Луну, однако в реальности состоялось только пять («Аполлон-13, -14, -15, -16, -17»). В период с 1978 по 1980 год планировалось построить обитаемую станцию на окололунной орбите, а с 1980 по 1983 год — развернуть постоянную базу на поверхности Луны, однако эти два проекта остаются мечтой фантастов (рис.1).

1. ***Лунная база «Звезда».***

В документах ГСКБ Спецмаш проект проходил под обозначением «ДЛБ» («Долговременная лунная база»), в ОКБ-1 его знали под красивым названием «Звезда» (рис.2). В конструкторском бюро Спецмаша изучался самый широкий круг вопросов, связанных с лунной базой (цели базы, принципы строительства, стадии развертывания и состав научного и строительного оборудования). Для решения ряда проблем приходилось привлекать смежников из других организаций.

Предполагалось, что место для базы будет выбрано с использованием автоматических аппаратов. С орбитального спутника Луны будет произведено картографирование участка, затем беспилотная станция возьмет пробы грунта и доставит их на Землю, после этого район будущего строительства обследуют луноходы. По окончании этапа дистанционного изучения предполагаемой территории базы на Луну отправится экспедиция из четырех человек на «лунном поезде».

«Лунный поезд» конструкции КБ Бармина предназначался для строительства временного городка, а по его завершении — для научных вояжей по окрестностям. В него входили: тягач, жилой вагончик, изотопная энергоустановка мощностью 10 кВт и буровая установка. Ходовая часть у всех этих машин была, как у луноходов: каждое колесо имело свой электромотор, благодаря чему отказ одного или даже нескольких из 22 моторов не парализует общий ход.

Главной задачей экипажа «лунного поезда» должны были стать геологические исследования: сначала - для подбора участков под городок и космодром, потом - для решения научных вопросов.

Основными проблемами создания лунных баз являются:

* Доставка модулей базы
* Сборка модулей базы
* Проблемы энергетики
* Защита от радиации

Очевидно, что создание подобного рода баз - комплексная и на настоящий момент нерешенная задача, основные сложности которой связаны с экстремальными условиями работы в открытом космосе, где работа человека ограничена или полностью невозможна.

Одним из перспективных способов построения лунной базы является использование группы роботов, осуществляющих строение инфраструктуры в автономном режиме. Данные аппараты смогут не только осуществлять сборку модулей, но и строить активные сооружения защиты от радиации из реголитных блоков. Иллюстрация возможного примера работы представлена на рис.3.

Группа роботов способна выполнять поставленные задачи самостоятельно, без внешних управляющих воздействий путем взаимодействия друг с другом и окружающей средой в разы быстрее и эффективнее по сравнению с единичным аппаратом.

Важным фактором успешного коллективного взаимодействия автономных аппаратов является их информационная осведомленность, для чего используются сложные системы технического зрения. Самой трудоемкой задачей является анализ изображения и формирование управляющего воздействия.

**Постановка задачи**

Цель работы - разработка алгоритмов распознавания объектов и определения их характеристик для построения инфраструктуры группой роботов в автономном режиме.

Для достижения поставленной задачи необходимо:

* Провести анализ возможных способов распознавания объектов, а так же методов улучшения качества и скорости распознавания;
* Сформировать этапы обработки информации с камеры;
* Разработать программное обеспечение в соответствии с предложенными этапами обработки;
* Протестировать разработанное программное обеспечение;

**Разработка алгоритмов**

Существует много различных методов распознавания объектов, но пока универсального алгоритма для решения этой задачи не существует, и каждое конкретное задание требует тщательного анализа и выбора особого подхода к его решению в зависимости от различных условий работы и окружающей среды.

**Этапы цифровой обработки изображений**

****

После получения изображения требуется устранить шумы и искажения для эффективной работы алгоритмов, далее выделить признаки и границы объектов, а затем применить способы детектирования.

В работе рассмотрены некоторые из возможных способов детектирования плоских фигур, а также их объединение для получения наилучшего результата.

Перед определением объекта необходимо получить внешние очертания объекта, то есть его контур.

При обнаружении контуров внутренние точки объекта во внимание не принимаются.

При выполнении контурного анализа возникают следующие проблемы:

* из-за одинаковой яркости с фоном объект может не иметь чёткой границы, или может быть зашумлён помехами, что приводит к невозможности выделения контура;
* перекрытие объекта приводит к тому, что контур выделяется неправильно и не соответствует границам объекта.

Несмотря на вышеперечисленные недостатки, переход от анализа пространства изображения к анализу пространства контуров существенно ускоряет вычисления.

Я отобрал наиболее эффективные методы получения контуров объектов

* *Детектор границ Кэнни*

Является одним из самых эффективных способов выделения границ. При анализе изображения краями являются кривые, вдоль которых наблюдаются резкие переходы яркости. Причиной является изменение освещенности, цвета, ориентации поверхности.

При преобразовании изображения в набор кривых сокращается объем информации для дальнейшего анализа, а также основные характеристики изображения. Работает только с изображениями в оттенках серого.

Алгоритм детектора:

* + - Убрать шум и лишние детали из изображения;
    - Рассчитать градиент изображения;
    - Уменьшить толщину краев;
    - Связать края в контуры

Так как детектор использует гауссовый фильтр, он неустойчив к помехам, и применять его следует только после фильтрации изображения.

* *Пороговое преобразование*

Изображение представляет собой массив пикселей. Установив пороговое значение, можно получить существенные характеристики изображения.

Обычное пороговое преобразование никак не учитывает, что части объектов могут иметь различную яркость из-за разности в освещённости. Это можно исправить, если использовать адаптивное пороговое преобразование, которое рассматривает значение не в одном пикселе, а в окрестности пикселя. Это значение может быть просто средним значением пикселей окрестности(т.е. все пиксели равнозначны), либо пиксели окрестности умножаются на весовой коэффициент (взвешиваются) в соответствии с функцией, например, с гауссовой функцией.

Пример практического применения этих функций — выборка определённых областей изображения, которые отличаются от других яркостью.

* *Выделение объекта по цвету (HSV)*

**HSV** ( (рис. 4 ) Hue / Saturation / Value - тон, насыщенность, значение) - цветовая модель, в которой координатами цвета являются:

* + - **Hue -** цветовой тон. Варьируется в пределах 0—360°, иногда приводится к диапазону 0-100 или 0-1.
    - **Saturation** - насыщенность. Варьируется в пределах 0-100 или 0-1.
    - **Value -** значение цвета или яркость. Также задаётся в пределах 0-100 и

0-1.

Данный способ обнаружения очень удобен при поиске одного или нескольких объектов, имеющих необходимый цвет или метку для обнаружения.

После получения контура необходимо хранить полученные данные для дальнейшего обращения к ним. Один тип объектов, которые могут храниться в хранилище памяти - это последовательность. Сами по себе, последовательности являются связанными списками других структур. Последовательность - это двусторонняя (дек)

Далее будут показаны несколько методов распознавания на примере плоских фигур: окружность, четырехугольник, треугольник.

Алгоритм №1.

Алгоритм, основанный на вычислении отношения площади к периметру, прост в реализации и в большинстве случаев дает верный результат. После получения контура с помощью детектора границ Кэнни, объект определяется с помощью функции, где текущий контур объекта проверяется с помощью приведенного ниже условия и выделяется цветом для наглядности, написанного на языке C++:

//определение окружности

if ( area / (perim \* perim) > 0.07 && area / (perim \* perim)< 0.1){

cvDrawContours(\_image, current, cvScalar(0, 0, 255), cvScalar(0, 255, 0), -1, 20, 20);

//рисуем контур

}

Получив периметр и площадь контура, с помощью дополнительных коэффициентов (подобранных для каждой фигуры) и порогового значения можно определить, какой фигурой является данный объект.

Такой алгоритм не устойчив к искажениям и шумам. Также стоит отметить, что коэффициенты подлежат изменению даже при небольших изменениях входных данных.

На рис.5. представлен пример работы этого алгоритма , настроенного на поиск прямоугольников. У многих кирпичей присутствуют искривления контура, поэтому только часть объектов распознается как прямоугольники.

Алгоритм №2

Полученный алгоритм различает фигуры по количеству вершин и типу углов. Этот алгоритм работает гораздо быстрее и точнее предыдущего. После получения контура проверяется количество вершин.

Пример условия для определения четырехугольника на языке c++:

result - хранилище вершин для конкретного контура

total – хранилище вершин, получаемое с помощью Рагера - Дугласа – Пекера( аппроксимация кривой с настраиваемой точностью)

if( result ->total==4 )//если количество вершин равно 4

{

//получение координат из хранилища контуров

CvPoint \*pt[4];

for(inti=0;i<4;i++){

pt[i] = (CvPoint\*)cvGetSeqElem(result, i);

}

//выделяем контур четырехугольника

cvLine(src, \*pt[0], \*pt[1], cvScalar(0,0,250),4);

cvLine(src, \*pt[1], \*pt[2], cvScalar(0,0,250),4);

cvLine(src, \*pt[2], \*pt[3], cvScalar(0,0,250),4);

cvLine(src, \*pt[3], \*pt[0], cvScalar(0,0,250),4);

}

Однако он имеет минусы: при близком расположении фигур друг к другу алгоритм может путать контуры, результат работы зависит от исправности контура.

Алгоритм №2 справляется лучше Алгоритма №1, но заметны помехи, вызванные неровностью контуров

(рис. 6).

Данный алгоритм был применен к другому изображению и тоже показал хороший результат. Работа алгоритма проиллюстрирована на рис. 7.

Алгоритм №2 был применен к реальным изображениям, полученным с камеры.

Результаты работы алгоритма изображены на рис.8,9.

Полученные алгоритмы эффективны лишь на «чистых» изображениях и не учитывают искажений, что негативно влияет на результат, поэтому для повышения качества я применил фильтрацию.

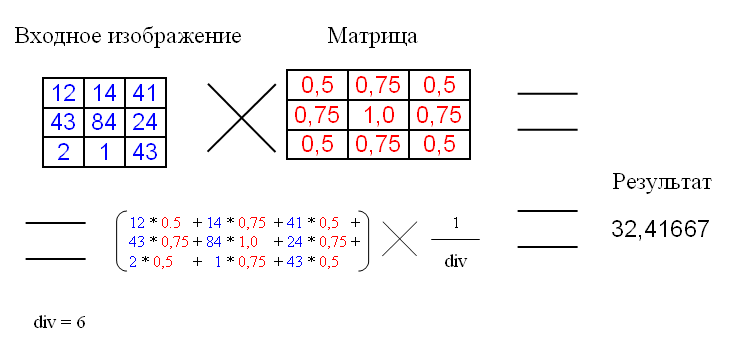
Устранение шумов

Эффективность вышеперечисленных способов сильно зависит от зашумленности, поэтому важно устранять шумы и искривления на входном изображении. Я проанализировал разные способы устранения оптических искривлений и выбрал наиболее оптимальные.

* Медианный фильтр – изменение каждого пикселя на среднее значения соседних пикселей;

Для текущего пикселя, пиксели, которые попадают в матрицу, сортируются, выбирается среднее значение из отсортированного массива. Это значение и является выходным для текущего пикселя.

Фильтр работает с матрицами различного размера и размер матрицы влияет только на количество рассматриваемых пикселей



* Сглаживание по гауссиане в области 3 на 3;

Для двух измерений формула будет выглядеть так

C:\Users\user\Desktop\1fafabfdfb9d1057c61a51d9dcf97ca958262aeb.png

Где \mu среднее значение и \sigma является отклонением (для каждогоx x и y)

* Cвёртка - операция вычисления нового значения заданного пикселя, в которой учитываются значения окружающих его соседних пикселей.   
  Главным элементом свёртки является так называемое  ядро свёртки — матрица (произвольного размера и отношения сторон; чаще всего используется квадратная матрица ( часто 3х3).

При вычислении нового значения выбранного пикселя изображения ядро свертки прикладывается своим центром к текущему пикселю. Соседние пикcели также накрываются ядром.

Далее вычисляется сумма произведения значений пикселей изображения на значения, данный пиксель элемент ядра.

Полученная сумма является новым значением выбранного пикселя.

После применения фильтрации качество распознавания объектов на предыдущих изображениях, полученных с камеры, улучшилось (рис.10. рис.11.).Это показывает необходимость ее применения.

Определение окружностей ведется с  помощью градиентного метода Хафа.

Каждый алгоритм имеет свое слабое место, поэтому применение комбинаций различных алгоритмов позволяет добиться максимально точного результата для заданных объектов инфраструктуры за кратчайшее вычислительное время, что очень важно при групповом управлении роботов. К примеру, на рис(10 и 9) для обнаружения окружностей используется преобразование Хафа, а остальные объекты определяются по количеству вершин для более быстрого получения результата, ядро свертки автоматически настраивается в зависимости от входящих данных.

Преобразование Хафа основывается на представлении объекта в виде параметрического уравнения. Параметры этого уравнения представляют собой фазовое пространство. После получения двоичного изображения перебираются все точки границ и делается предположение, что точка принадлежит линии искомого объекта. Финальным шагом является обход пространства Хафа и выбор максимальных значений, за которые проголосовало больше всего пикселей картинки, что и дает параметры лдя уравнений искомого объекта.

Алгоритм метода:

* + - используется детектор границ Кенни для нахождения границ на изображении;
    - для ненулевых точек высчитывается градиент;
    - определяются центры кругов;
    - относительно центра определяются ненулевые точки, лежащие на одном расстоянии;

Работает с 8 битными одноканальными изображениями в оттенках серого.

Использование моментов для нахождения центра масс

Нахождение центра масс необходимо для эффективного перемещение объекта в пространстве. Центр масс определяется с помощью моментов.

Момент — это суммарная характеристика контура, рассчитанная интегрированием всех пикселей контура. Моменты очень удобны для описания объектов после сегментации, а так же для поиска по шаблону.

момент (p,q) определяется формулой:

 Где p и q — порядок возведения в степень соответствующего параметра при суммировании,n — число пикселей контура.

Однако такая формула имеет недостатки:

* отсутствие возможности сравнения двух контуров одинаковых фигур разных размеров;
* зависимость от координат, что не позволяет определить фигуру под углом.

Для устранения этих недостатков было решено использовать нормализованные инвариантные моменты.

Где Xc и Yc– центр масс

Сам центр масс является средним взвешенным пикселей объекта, где вес является интенсивностью, найти его можно через отношение моментов.

Данное фото демонстрирует результат работы текущей версии программы: возможность определять заданные объекты (четырехугольник и его тип, окружность, треугольник) и находить их центр тяжести (рис.12).(К работе приложен gif файл, демонстрирующий работу программы в реальном времени)

**Заключение**

Результаты работы - проведен анализ наиболее оптимальных методов детектирования, разработаны и протестированы различные алгоритмы распознавания объектов и определения их характеристик, проводится разработка группового управления роботами. Полученные в ходе работы алгоритмы способны работать в режиме реального времени с высокой точностью.

Планируется создание обучаемой модели на основе полученных способов детектирования, совмещение полученных алгоритмов с манипуляторами для испытания на стенде.

**Список Литературы**

1. Гонсалес Р. Цифровая обработка изображений в среде MATLAB. //«РИЦ»« Техносфера».2006
2. Документации библиотеки openCV, версия 2.4.13URL: <http://docs.opencv.org/2.4.13>
3. Свободная энциклопедия «Википедия» [Сайт]. Режим доступа: https://ru.wikipedia.org/wiki (Дата обращения 31.12.2016).

**Приложение**



Рис.1 . Лунная колония программы «Apollo»

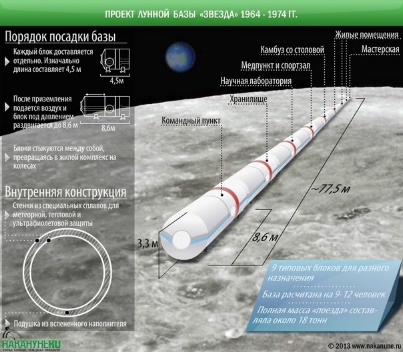


Рис. 2 Лунная база «Звезда»

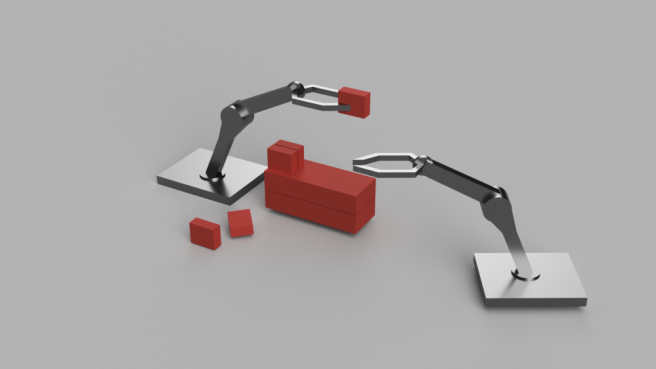


Рис.3.Иллюстрация работы группы роботов-манипуляторов.

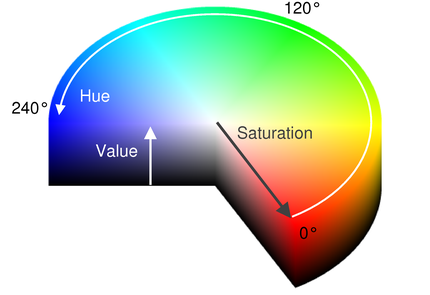


Рис.4. **HSV**  (рис. 4)(Hue, Saturation, Value — тон, насыщенность, значение) — цветовая модель

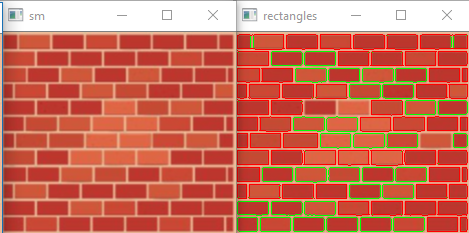
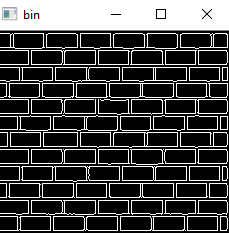


Рис.5..Работа алгоритма №1 а) исходное изображение б) полученные контуры, с) результат работы

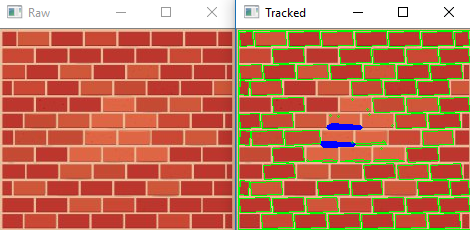


Рис.6. а) исходное изображение б) полученный результат

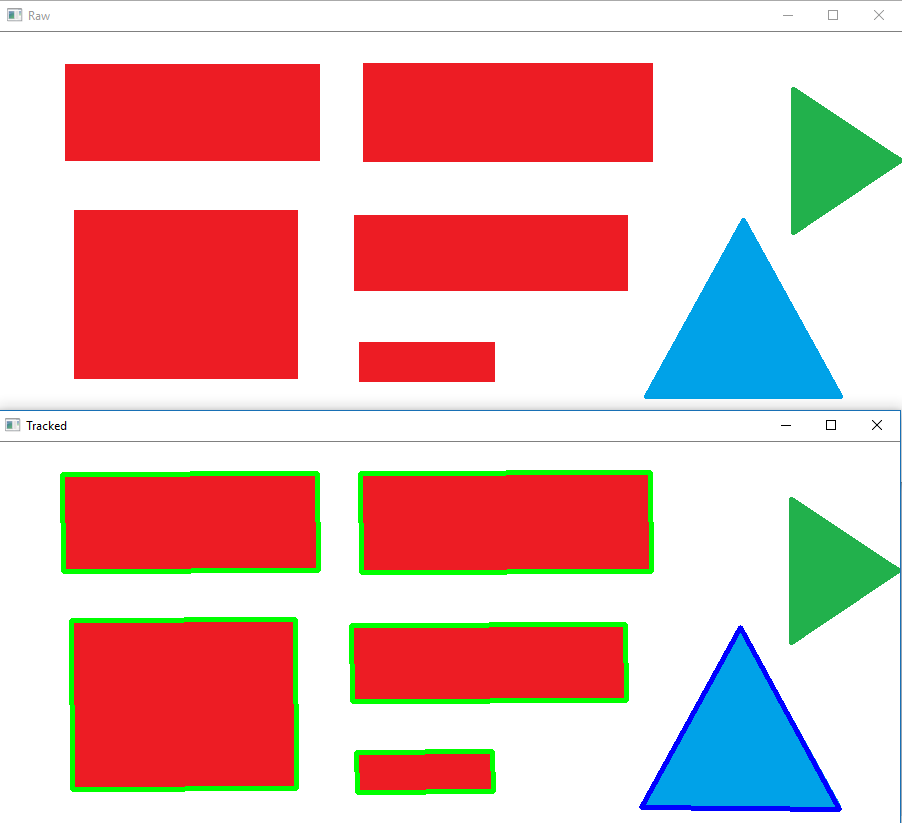


Рис.7. а) исходное изображение

б) полученный результат



Рис.8. Результат работы алгоритма № 2

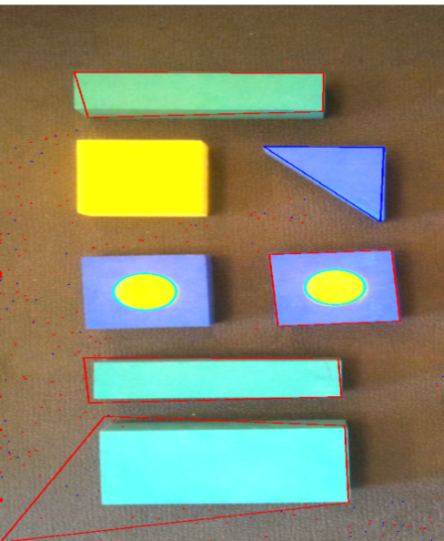


Рис. 9 . Результат работы алгоритма № 2

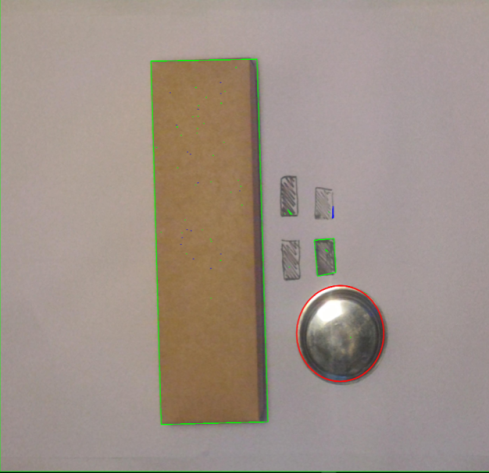
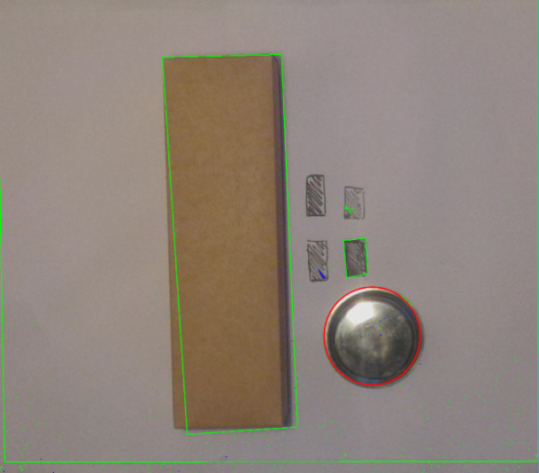


Рис.10. а) изображения до фильтрации б)после фильтрации

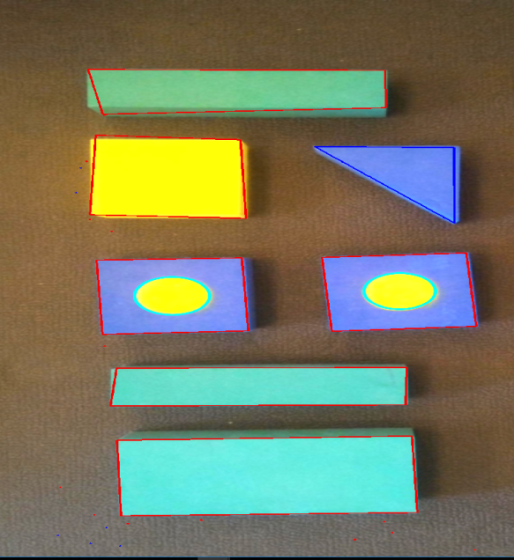
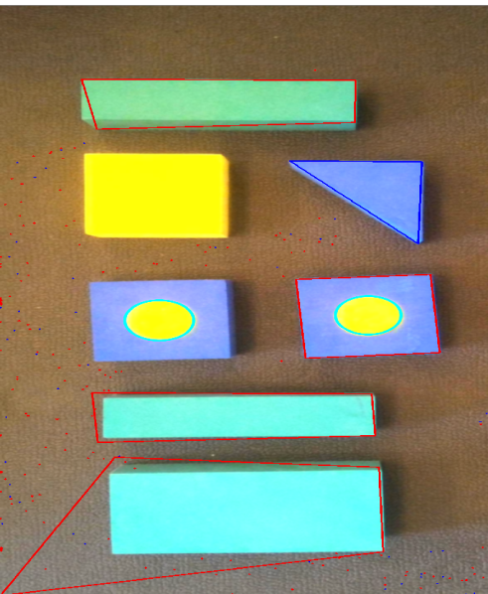


Рис.11. а) изображения до фильтрации б)после фильтрации

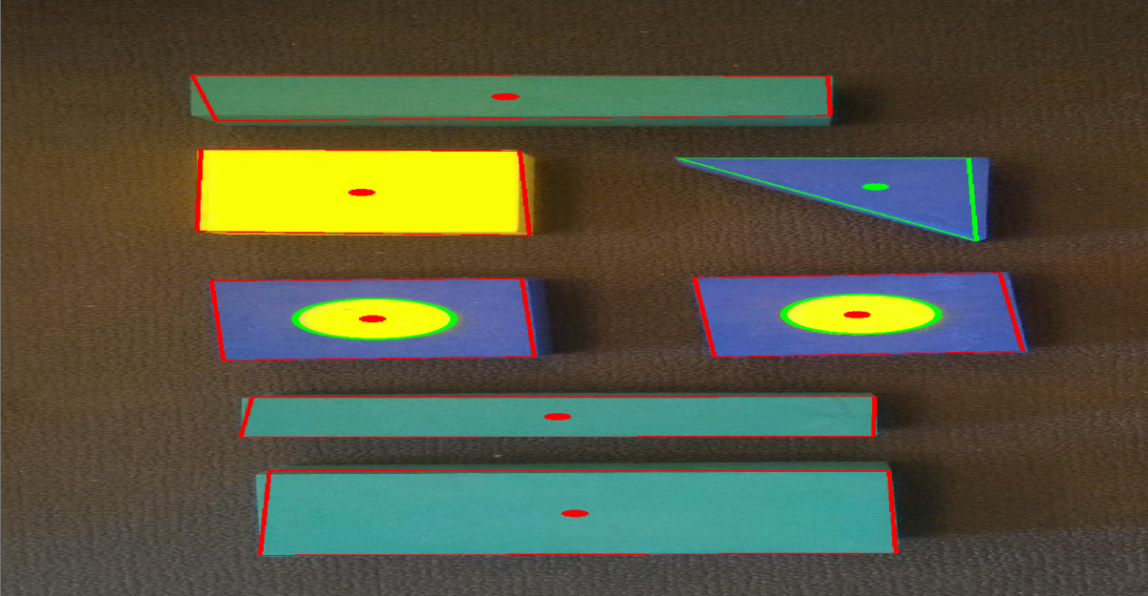


Рис.12.Результат работы текущей версии программы: