

Содержание

Введение.....	2
1. Реализация макета системы.....	3
2. Обоснование выбора структуры системы.....	7
3. Подбор элементной базы	7
4. Обеспечение детектирования объекта по цвету	9
4.1. Избавление от ненужных объектов.....	12
4.2. Исследование зависимости точности определения от освещения	13
4.3. Построение интерфейса выбора цвета и условий освещенности	16
4.4. Реализация вычисления необходимого угла поворота	17
Заключение	19

Введение

Основную часть информации о внешнем мире человек получает по зрительному каналу и далее весьма эффективно обрабатывает полученную информацию при помощи аппарата анализа и интерпретации визуальной информации. Поэтому стоит вопрос о возможности машинной реализации данного процесса.

За счет возрастания сложности решаемых научно-технических задач, автоматическая обработка и анализ визуальной информации становятся все более актуальными вопросами. Данные технологии используются в весьма востребованных областях науки и техники, таких как автоматизация процессов, повышение производительности, повышение качества выпускаемых изделий, контроль производственного оборудования, интеллектуальные робототехнические комплексы, системы управления движущимися аппаратами, биомедицинские исследования и множество других. Кроме того, можно сказать, что успех современного бизнеса основывается главным образом на качестве предлагаемой продукции. А для его обеспечения, если говорить о производстве материальных вещей, требуется визуальный контроль.

Далее мы будем использовать термин «машинное зрение» (Machine vision) как понятие, наиболее полно охватывающее круг инженерных технологий, методов и алгоритмов, связанных с задачей интерпретации визуальной информации, а также как практическое использование результатов этой интерпретации.

В настоящее время на производстве случается достаточное количество браков, таких как: искривление радиуса у подшипников, неравномерная покраска деталей автомобиля, не соответствие заданных параметров какого-либо изделия и т.д.

Целью данной работы является попытка отследить брак и по

возможности предотвратить их появление на производстве.

Область применения данной разработки весьма обширна, т.к. брак может случиться на любом производстве. Возьмем к примеру покрасочный манипулятор кузовных деталей автомобиля. Покрасочный материал может лечь неравномерно на деталь автомобиля и с помощью технического зрения данной системы можно отследить какую именно деталь стоит перекрасить по второму слою.

Для достижения цели выпускной квалификационной работы, создания алгоритма по распознаванию цветного объекта, были поставлены следующие задачи:

- 1) разработка технического задания;
- 2) изучение основ машинного зрения;
- 3) выбор структуры системы;
- 4) подбор элементной базы;
- 5) настройка проекта Visual studio для работы с OpenCV;
- 6) реализация детектирования объекта по цвету;
- 7) реализация детектирования объекта по форме;
- 8) реализация калибровки;
- 9) реализация интерфейса пользователя с возможностью выбора цвета и условий освещенности;

1. Реализация макета системы.

Для создания макета был использован инновационный инструмент проектирования AutoCAD. Данный макет был разработан нидерландским аспирантом в 2013 году для реализации задачи прямого управления манипулятора. Было решено создать свой проект на базе данной разработки с последующей ее модификацией.

Все готовые чертежи были сконвертированы в машинный код с помощью программы MakerBot для дальнейшей реализации на 3D – принтере. Интерфейс программы представлен на рисунке 1.

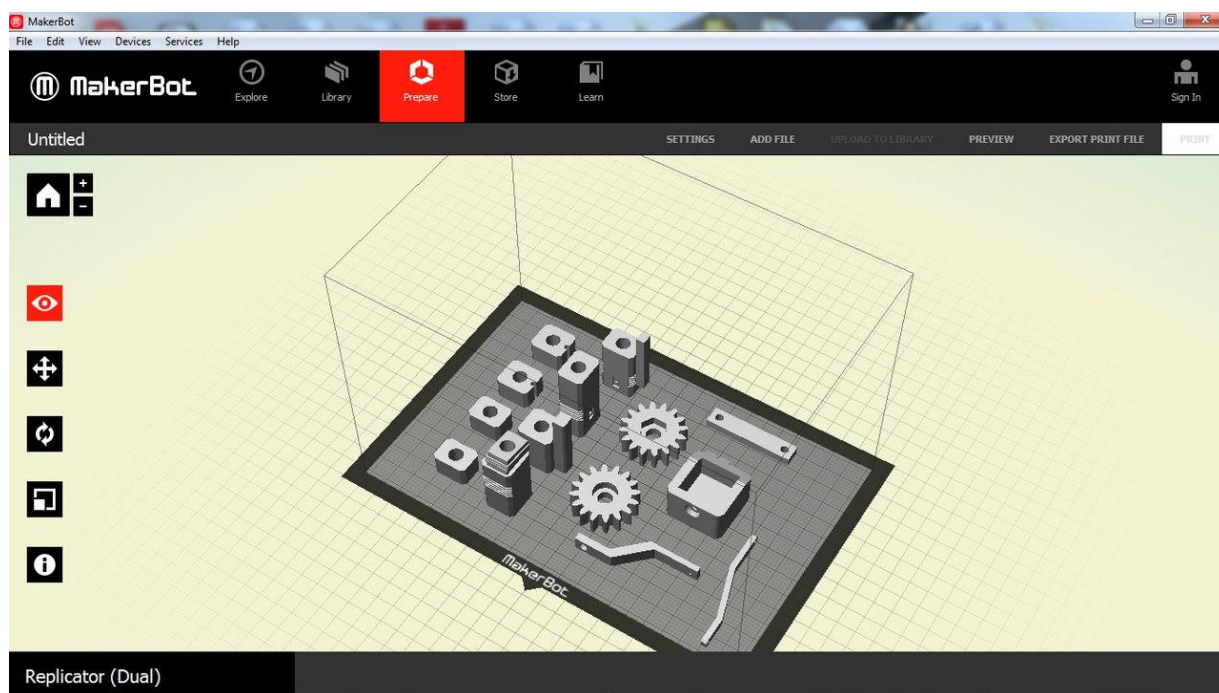


Рисунок 1. Интерфейс вспомогательной программы MakerBot.

Было напечатано 29 деталей и использовано 3 килограмма полимера для 3D – принтера. Напечатанные детали представлены на рисунке 2.



Рисунок 2. Напечатанные детали манипулятора.

Использовался 3D – принтер компании Wanhao модели Duplicator 4 из-за его дешевизны и надежности. На рисунке 3 изображен данный 3D – принтер.

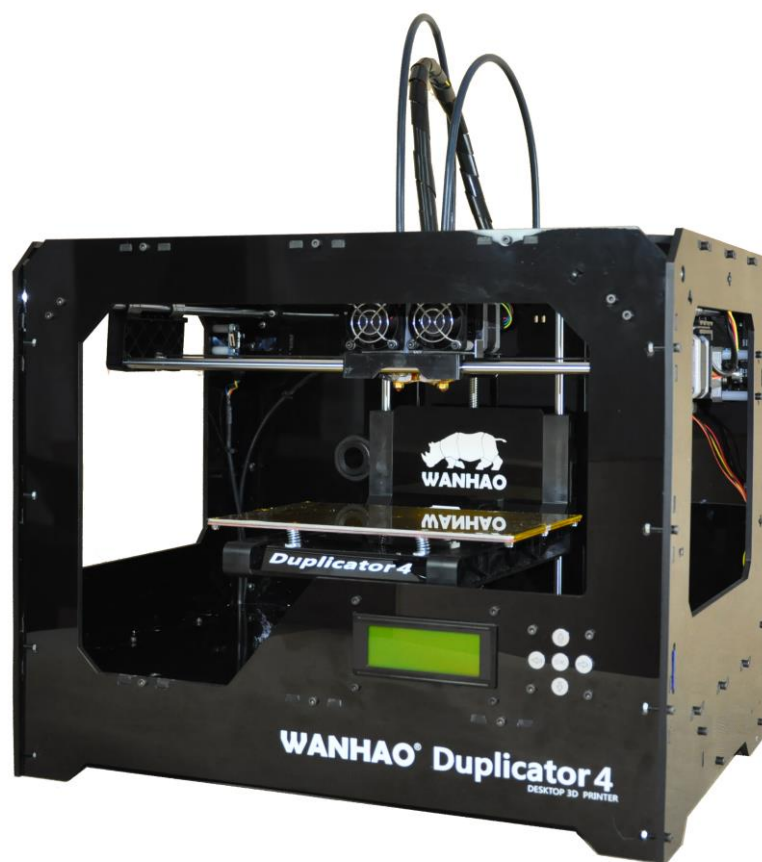


Рисунок 3. 3D – принтер модели Duplicator 4.

Модель данной системы представлена на рисунках 4, 5, 6.



Рисунок 4. Виртуальная модель системы



Рисунок 5. Вид сбоку



Рисунок 6. Фронтальный вид

Вес данной модели составляет около двух килограмм, так как все детали напечатаны из пластика.

2. Обоснование выбора структуры системы

Выбор структуры системы обусловлен дальнейшим ее использованием в других задачах. В рамках данной выпускной квалификационной работы данная структура является излишне сложной, так как можно было обойтись только веб-камерой и аппаратной частью. В последующих запланированных работах будут реализованы задачи, использующие полный потенциал данной разработки.

3. Подбор элементной базы

Сервоприводы использовались цифровые. Внешне они мало чем отличаются от аналоговых, но основное отличие заключается в способе обработки сигнала приемника и управления двигателем. Рассмотрим графики зависимости вольтажа от времени на рисунке 7.

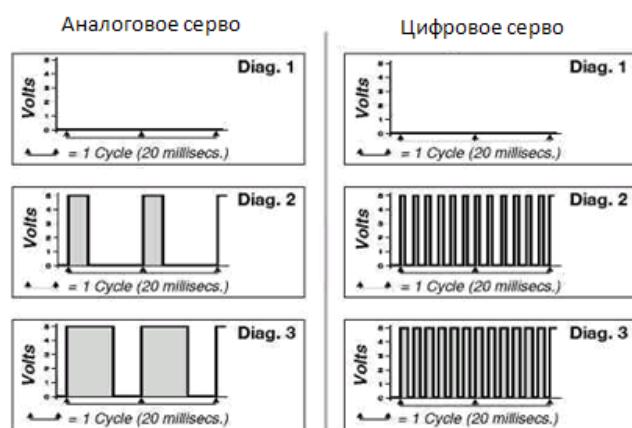


Рисунок 7. График зависимости вольтажа от времени

Сервоприводы использовались моделей sm-4209m и sm-s2213m. Технические характеристики данных сервоприводов:

sm-4209m

Характеристики:

Размер: 42.2x20.5x39.5 мм
 Усилие на валу: 0,087 кг/м (4.8V); 0,096 кг/м (6.0V)
 Скорость: 0.13 сек./60 град. (4.8v); 0.11 сек./60 град. (6.0v)
 Рабочее напряжение: 4.8V/6.

sm-s2213m

Характеристики:

Размер: 22.2x11.6x21.5 мм
 Усилие на валу: 0,0161 кг/м (4.8V); 0,018 кг/м (6.0V)
 Скорость: 0.09 сек./60 град. (4.8v); 0.08 сек./60 град. (6.0v)
 Рабочее напряжение: 4.8V/6.

Веб-камера использовалась с высоким разрешением в 720p, так как необходимо достаточно четкое изображение для получения бинарного изображения с последующим подсчетом пикселей. На рисунке 8 изображена данная веб-камера.



Рисунок 8. Веб-камера.

4. Обеспечение детектирования объекта по цвету

Способность распознавать объекты по их цвету и форме необходима для функционирования автономного робота. В целях экономии вычислительных ресурсов, робот может осуществлять движение по патрулируемой территории по цветным маркерам. Для обнаружения маркера робот должен определить его по цвету, а затем по форме, для того, чтобы не перепутать маркер с другими объектами того же или близкого к тому же цвета.

Как известно, любое изображение, представленное в цифровой кодировке RGB представляет собой совокупность трех цветов – красного (Red), зеленого (Green) и синего (Blue). Их соотношение в конкретном месте изображения будет определять его цвет.

Очевидно, что область красного цвета содержит в себе гораздо больше красной составляющей, чем остальных двух тоже самое справедливо и для

остальных цветов. Обобщенный алгоритм определения нужного цвета представлен на рисунке 9.

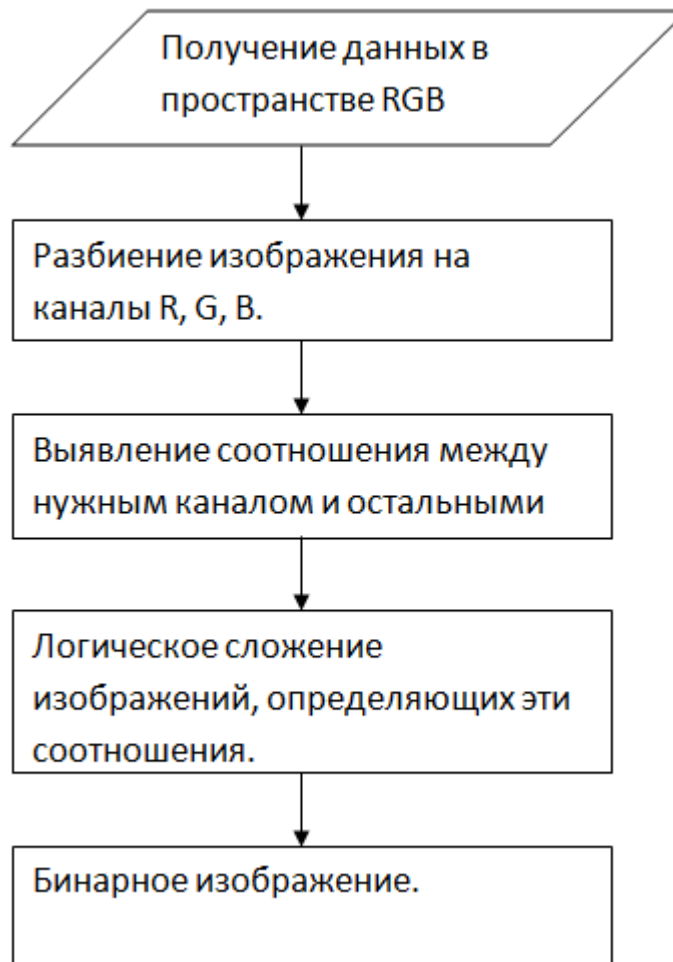


Рисунок 9. Алгоритм определения цвета

На программном уровне это выглядит следующим образом:

```
Mat video;  
  
while (true)  
{  
    capture.read(video);  
    imshow("out", video);  
  
    vector<Mat> planes;  
    split(video, planes);  
  
    Mat gray1 = planes[2] / planes[1];
```

```
Mat gray2 = planes[2] / planes[0];  
Mat gray = gray1 + gray2;
```

Поскольку в разных местах изображения соотношение каналов разное, то на выходе будет неоднородное изображение с переходящими оттенками серого, что затрудняет детектирование. Модифицируем алгоритм так, чтобы все пиксели подходящего нам цвета были белыми, а не подходящие – черными.

Для этого с помощью средств OpenCV произведем обход всех пикселей нашего изображения, сравним их значения с критерием, при котором можно считать цвет подходящим. Этот критерий показывает во сколько раз присутствие красного должно быть больше остальных. Если пиксель подходит, то присваиваем ему значение в 255, то есть, заливаем белым.

```
for (int y = 0; y < gray.rows; y++)  
{  
    for (int x = 0; x < gray.cols; x++)  
    {  
        int value = gray.at<uchar>(y, x);  
        if (value >= 3)  
        {  
            n = n + 1;  
            gray.at<uchar>(y, x) = 255;  
        }  
    }  
}
```

В результате на рисунке 10 виден объект, окрашенный в красный цвет.

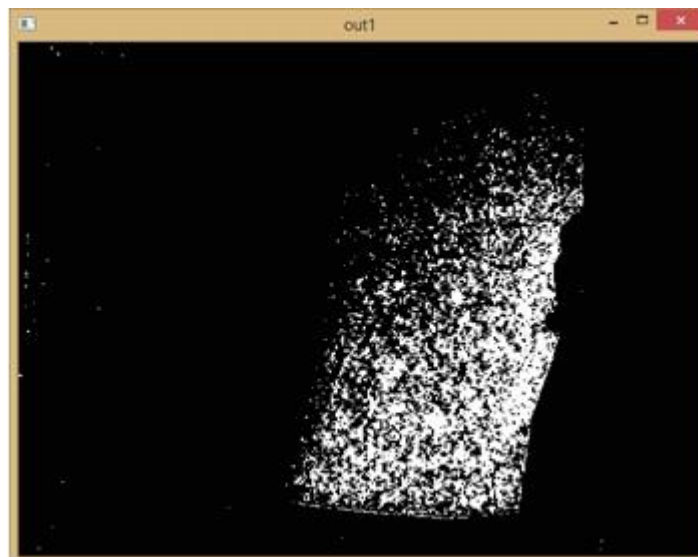


Рисунок 10. Видимый объект.

4.1. Избавление от ненужных объектов

Как показывает предыдущий пункт, мы можем определять нужные цвета, но сталкиваемся с большим количеством помех и мелких предметов, мешающих воспринимать объект.

Для того, минимизировать влияние таких явлений произведем Гауссово сглаживание нашего изображения – оно устранил помехи.

```
GaussianBlur(gray, gray, cv::Size(21, 21), 0);
```

В результате чего на рисунке 11 видно, что объект стал виден намного лучше.



Рисунок 11. Видимый объект.

4.2. Исследование зависимости точности определения от освещения

Как показывает практика, помехи – не единственная проблема, с которой сталкивается техническое зрение при определении объекта. Освещенность местности также серьезно влияет на точность измерений.

Это происходит из-за того, что в зависимости от количества света цвета приобретают ту или иную степень яркости, которая влияет на критерий соотношения, по которому цвет будет определяться.

При низком освещении этот критерий меньше, а при высоком должен быть больше.

Влияние света на детектирование при постоянном критерии соотношения между цветами проиллюстрировано на рисунках 12, 13, 14, 15.

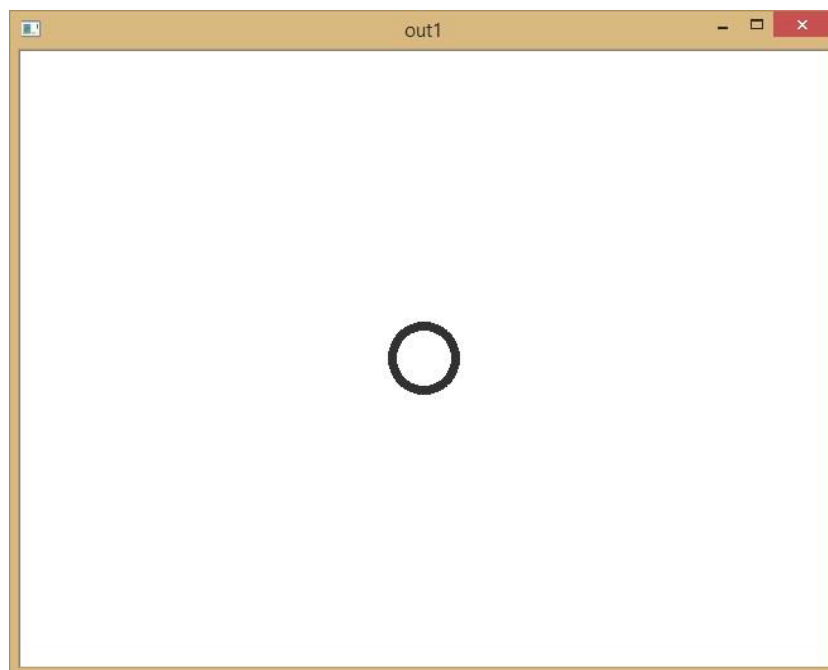


Рисунок 12. Очень высокий уровень освещенности.



Рисунок 13. Объект при нормальных условиях освещенности.



Рисунок 14. Объект при слабом освещении.



Рисунок 15. Объект при очень низком уровне освещенности.

4.3. Построение интерфейса выбора цвета и условий освещенности

Как показало исследование, систему нужно адаптировать к разным условиям освещенности для обеспечения ее стабильного функционирования.

Для этого был создан интерфейс, позволяющий пользователю в режиме реального времени настраивать условия освещенности.

Как уже было выяснено, количество света главным образом влияет на коэффициент соотношения между цветами, следовательно, необходимо создать возможность выбора его значений.

Для этого заменим коэффициент соотношения переменных и используем встроенную в OpenCV функцию `TrackBar` для создания ползунков, меняющих значения переменных $B_{\min, \max}$, $R_{\min, \max}$, $G_{\min, \max}$:

Также должна быть предусмотрена возможность гибкого использования системы для опознавания нескольких цветов, а не только красного. Также используя `TrackBar` создадим несколько возможных ситуаций, соответствующих красному, синему и зеленому цветам:

В итоге имеем систему, которая может различать объекты трех основных цветов:

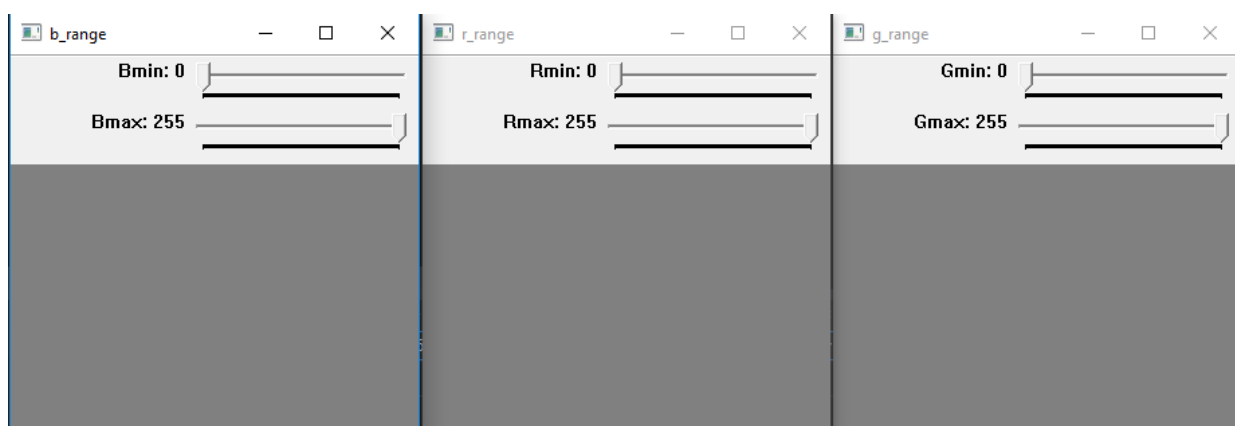


Рисунок 16. RGB - ползунки.

4.4. Реализация вычисления необходимого угла поворота

Чтобы система могла следить за объектом с помощью сервопривода, необходимо знать угол отклонения объекта от центральной оси кадра камеры.

Следить за всем объектом не удобно, поэтому вычислим точку – центр масс этого объекта, за которой и будет вестись наблюдение. Координата центра масс будет соответствовать сумме значений всех координат пикселей объекта деленное на количество этих пикселей.

Вычислим необходимые параметры, введя в систему два счетчика, обозначим точку слежения маркером.

```
for (int y = 0; y < rgb_and.rows; y++)

    for (int x = 0; x < rgb_and.cols; x++)
    {
        uchar color = rgb_and.at<uchar>(Point(x, y));

        if (color == 255)
        {
            total_x += x;
            total_y += y;
            counter++;
        }
    }
```

На рисунке 17 показан центр масс данного объекта.

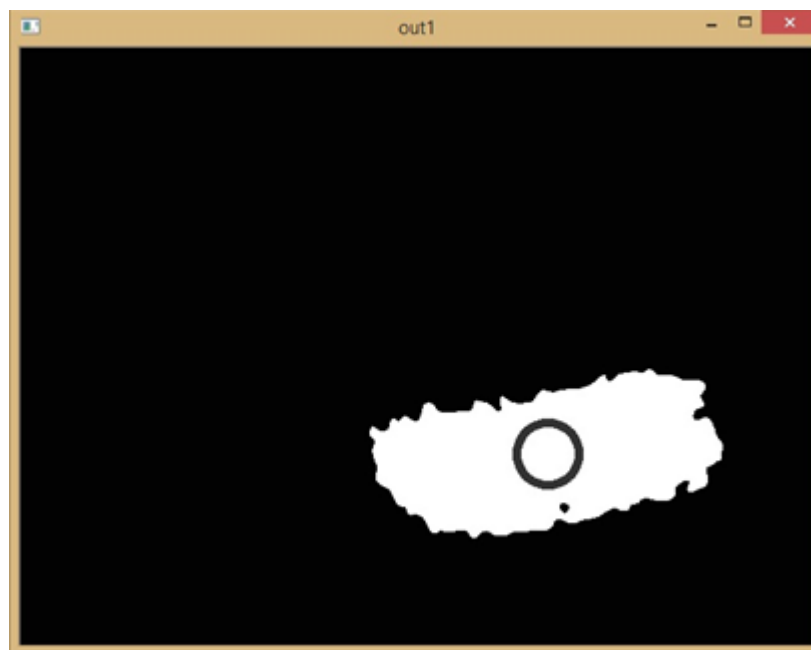


Рисунок 17. Центр масс объекта

Заключение

В ходе выполнения данной выпускной квалификационной работы были достигнуты следующие результаты:

- 1) составлено техническое задание, устанавливающее основные требования и условия эксплуатации устройства;
- 2) спроектирован макет системы;
- 3) написан алгоритм детектирования цветного объекта.

Таким образом была разработана система отслеживания производственных браков на примере распознавания цвета.