

## ПРОГРАММА ПОИСКА ОПРЕДЕЛЕННОГО ОБЪЕКТА ИЗОБРАЖЕНИЯ НА ОСНОВЕ ЕГО КОНТУРНОГО АНАЛИЗА

Ван Шии, Цапко И.В.

Томский политехнический университет  
729810575@qq.com

### Введение

В последнее время широкое применение при обработке изображений находит дисциплина компьютерного зрения, которая может создавать системы для получения разной информации о представленных объектах.

При этом, исходные данные могут быть получены из различных источников, например, фотоаппараты и видеокамеры, медицинские сканеры или устройства для сканирования трёхмерных объектов.

### Применения компьютерного зрения

Дисциплина компьютерного зрения нашла применение во многих отраслях человеческой жизни [1]:

1. В медицине, где видео- или фото-данные обрабатываются с целью получения из них информации для дальнейшей постановки диагноза больному. Как правило, исходные данные получают с помощью микроскопии, рентгенографии, ультразвуковых исследований и.т.д. Таким образом может быть получена информация о наличии опухолевых образований, атеросклероза или других изменениях в организме пациента.

2. В военной области для обнаружения неприятельских солдат и транспортных средств, а также при управлении ракетами. При этом ракета посылается в заданную область, а определение цели производится в момент, когда ракета достигает заданной области, основываясь на получаемых видеоданных.

3. Автономные транспортные машины. Уровень автономности изменяется от полностью автономных до транспортных машин, где системы, основанные на компьютерном зрении, поддерживают водителя в различных ситуациях. Полностью автономные транспортные машины используют компьютерное зрение для навигации, для получения информации, для создания карты, для обнаружения препятствий.

Типичными задачами компьютерного зрения являются распознавание объектов, определение динамических объектов и восстановление изображения.

Распознавание является классической задачей в компьютерном зрении, обработке изображений и машинном зрении. Данная задача в общем случае, а именно нахождение случайного объекта в случайной ситуации является нерешенной. В частных случаях распознавание позволяет определить содержат ли видеоданные некоторый характерный объект, особенность или активность.

Определение объектов, находящихся в динамике, связано с оценкой движения, в которых последовательность изображений обрабатываются для нахождения оценки скорости каждой точки изображения или трехмерной сцены [2].

Задача восстановления позволяет воссоздать трехмерную модель объекта или сцены.

### Алгоритм и программа поиска объекта

Данная работа посвящена разработке алгоритма и созданию программы для поиска определенных объектов на томографическом изображении (рисунок 1).

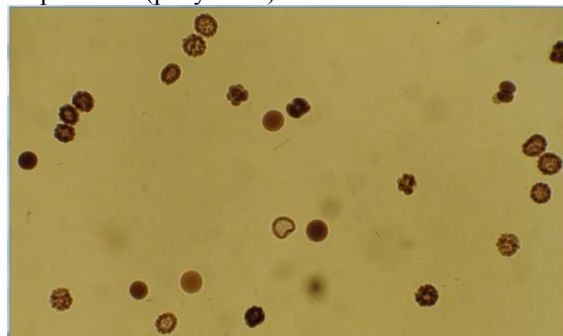


Рисунок 1- Исходное изображение

Программа разрабатывалась в среде Visual Studio 2013 на языке программирования C++ с использованием библиотеки компьютерного зрения OpenCV [3].

OpenCV является библиотекой компьютерного зрения с открытым исходным кодом. Она разработана компанией Intel на языке программирования C/C++.

Поиск определенных объектов, а именно круглых протопластов, на изображении проводился методом контурного анализа. Контур объекта – это его внешние очертания.

Как правило, при проведении контурного анализа полагается, что контур содержит достаточную информацию о форме объекта, при этом, внутренние точки во внимание не принимаются.

Последовательность действий при распознавании объектов методом контурного анализа включает в себя следующие шаги:

1. Предварительная фильтрация исходного изображения для устранения помех;
2. Бинаризация изображения;
3. Выделение контуров объектов;
4. Перебор всех найденных контуров для поиска максимально подходящего под шаблон.

Предварительная обработка изображения осуществляется с использованием операции свертки. Данная операция позволяет вычислить новое значение заданного пикселя на изображении с учетом значений окружающих его соседних пикселей.

В OpenCV операция свертки реализуется функцией:

```
CVAPI(void) cvFilter2D(const CvArr*  
src, CvArr* dst, const CvMat*  
kernel, CvPoint anchor  
CV_DEFAULT(cvPoint(-1,-1)));
```

где src — исходное изображение;  
dst — изображение для сохранения результата;  
kernel — ядро свертки;

anchor — якорь ядра (значение (-1,-1) говорит о том, что якорь находится в центре).

В качестве ядра свертки принимается квадратная матрица размером 3x3, позволяющая сглаживать исходное изображение.

Следующим этапом осуществляется двоичное преобразование изображения для выделения пикселей, лежащих в заданном интервале значений. Для этого используется функция:

```
CVAPI(void) cvInRangeS(const CvArr*  
src, CvScalar lower, CvScalar upper,  
CvArr* dst);
```

где src — исходный массив;  
lower — скаляр с нижней границей (включая);  
upper — скаляр с верхней границей (не включая);

dst — массив для хранения результата.

Для поиска контуров можно использовать функцию:

```
CVAPI(int) cvFindContours(CvArr*  
image, CvMemStorage* storage,  
CvSeq** first_contour, int  
header_size  
CV_DEFAULT(sizeof(CvContour)), int  
mode CV_DEFAULT(CV_RETR_LIST), int  
method  
CV_DEFAULT(CV_CHAIN_APPROX_SIMPLE),  
CvPoint offset  
CV_DEFAULT(cvPoint(0,0)));
```

где image — исходное 8-битное одноканальное изображение (ненулевые пиксели обрабатываются как 1, а нулевые — 0);

storage — хранилище памяти для хранения данных найденных контуров;

first\_contour — указатель, который будет указывать на первый элемент последовательности, содержащей данные найденных контуров;

header\_size — размер заголовка элемента последовательности;

mode — режим поиска.

Для данной работы, был выбран режим, позволяющий найти только крайние внешние контуры.

После определения контуров на изображении для каждого из них были вычислены площадь (количество пикселей в области) и длина

(количество пикселей на границе области). Последующее нахождение отношения этих величин позволяет определить компактность фигуры, ограниченной контуром.

Для получения значения инвариантного относительно радиуса разделим площадь круга на квадрат длины окружности:

$$\frac{\pi R^2}{(2\pi R)^2} = \frac{1}{4\pi} \sim 0.079577$$

Таким образом, можно определять круглые протопласты на исходном изображении.

В результате работы программы была проведена обработка исходного изображения (рисунок 1). На рисунке 2 показан результат обработки – выделены круглые протопласты, а на рисунке 3 показано количество определенных круглых объектов.

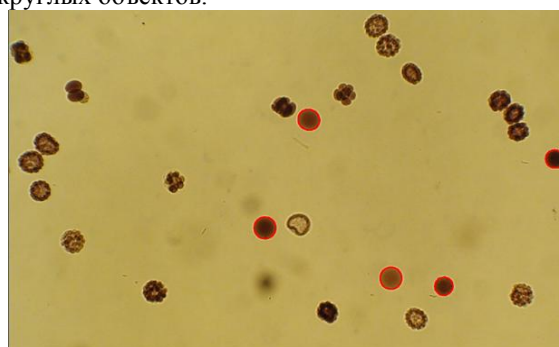


Рисунок 2 – Результат обработки

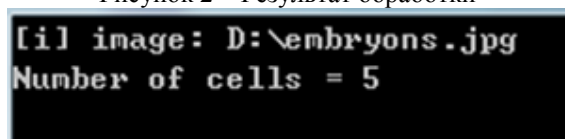


Рисунок 3 – Количество ячеек

### Заключение

В результате выполнения работы, определили количество круглых протопластов на исходном томографическом изображении.

На практике, познакомились с OpenCV, это библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Был изучен процесс создания программ в OpenCV, получены знания по операциям: сглаживание, изменение размеров, морфологические преобразования, поиск объекта по цвету, свертка, детектор границ Кенни (Canny) и т.д.

### Список использованных источников

1. Компьютерное зрение [Электронный ресурс]. – URL: <https://ru.wikipedia.org/> (дата обращения 28.09.2016).

2. Цапко И.В., Цапко С.Г., Гусакова Е.Г. Алгоритм создания трёхмерной сцены в 2DSMAX на основе двумерного снимка // Дизайн. Материалы. Технология. – 2012. № 4 (24). С. 107-112.

3. OpenCV шаг за шагом [Электронный ресурс]. – URL: <http://robocraft.ru/page/opencv/> (дата обращения 28.09.2016).