

ПРОГРАММА ДЕТЕКТИРОВАНИЯ ОБЪЕКТА ПО ШАБЛОНУ НА ИЗОБРАЖЕНИИ

Янь Фэнхуа, Цапко И.В.

Томский политехнический университет
250435126@qq.com

Введение

Компьютерное зрение опирается на информацию с видеокамер. Это могут быть и стереокамеры, и многокамерные системы. Использование базы трёхмерных объектов и текстур позволяет «понять», с чем именно столкнулся робот. Он отмеряет расстояние до объекта «воспринимая» пространство и, ориентируясь в нём, выделяет препятствия в зоне движения.

Зрение помогает роботу не просто работать с предметами и обстановкой, но и распознавать людей, их лица и жесты, опираясь на специальные программы распознавания. Эти данные в свою очередь запускают определённые сценарии поведения.

Применения компьютерного зрения

Работа с медицинскими изображениями: создание программных систем, которые могут улучшать набор изображений, выявлять на них важные моменты или события либо визуализировать информацию, полученную из изображений.

Различные технические проверки, когда по изображениям объектов определяется, соответствуют ли объекты спецификации [1].

Интерпретация фотографий, сделанных со спутника, как в военных целях (например, может потребоваться программа, выявляющая интересные, с военной точки зрения, события в указанном регионе или определяющая вред, нанесенный в результате бомбардировки), так и в гражданских (какой урожай кукурузы будет в этом году? сколько осталось тропических лесов?) [2].

Библиотека компьютерного зрения

Для реализации программы на языке C++ была использована библиотека компьютерного зрения OpenCV.

OpenCV – самая популярная библиотека компьютерного зрения. Она написана на C/C++, ее исходный код открыт, библиотека включает более 1000 функций и алгоритмов. Она разрабатывается с 1998 г., сначала в компании Интел, теперь в Itseez при активном участии сообщества. О высокой популярности библиотеки свидетельствует количество загрузок, их более 6000000.

Существуют библиотеки, более продвинутые по функциональности, например, Halcon. Есть библиотеки более специализированные, делающие

акцент на какой-либо конкретной задаче, например, libmv. Однако OpenCV – самая большая библиотека по широте тематики.

Библиотека распространяется по лицензии BSD, что означает, что ее можно свободно и бесплатно использовать как в открытых проектах с открытым кодом, так и в закрытых, коммерческих проектах. Библиотеку не обязательно копировать целиком в свой проект, можно использовать куски кода. Единственное требование лицензии – наличие в сопровождающих материалах копии лицензии OpenCV.

Из-за либеральной лицензии библиотека используется многими компаниями, организациями, университетами, например, NVidia, Willow Garage, Intel, Google, Stanford University. Компании NVidia и WillowGarage частично спонсируют ее разработку [3].

Программа детектирования объектов

Детектирование объектов — поиск объекта по шаблону. Детектирование объекта по шаблону может пригодиться во множестве случаев. Самый простой пример — поиск заранее заданного объекта [4].

Данная работа посвящена поиску определенных букв на изображении (в частности, необходимо было в тексте определить буквы «а» и подсчитать их количество).

В OpenCV детектирование объектов путем сравнения шаблона и перекрывающегося окна на исходном изображении осуществляется при помощи функции `cvMatchTemplate()`:

```
CVAPI(void) cvMatchTemplate( const CvArr*  
image, const CvArr* templ, CvArr* result, int  
method );
```

где `image` — изображение для поиска;
`templ` — шаблон для поиска (не должен превышать исходное изображение и иметь тот же тип);

`result` — карта результата сравнения;
`method` — метод сравнения областей изображения, задается в виде перечисления:

```
enum{  
CV_TM_SQDIFF =0,  
CV_TM_SQDIFF_NORMED =1,  
CV_TM_CCORR =2,  
CV_TM_CCORR_NORMED =3,  
CV_TM_CCOEFF =4,
```

```
CV_TM_CCOEFF_NORMED =5  
};
```

Функцию поиска можно приблизительно представить следующим образом: изображение шаблона `templ` последовательно накладывается на исходное изображение `image` и между ними вычисляется корреляция, результат которой заносится в результирующее изображение `result`. При этом, корреляция между двумя изображениями определяется параметром `method`.

В данной работе была использована следующая формула расчёта корреляции:

`Method=CV_TM_SQDIFF`

И получил результат.

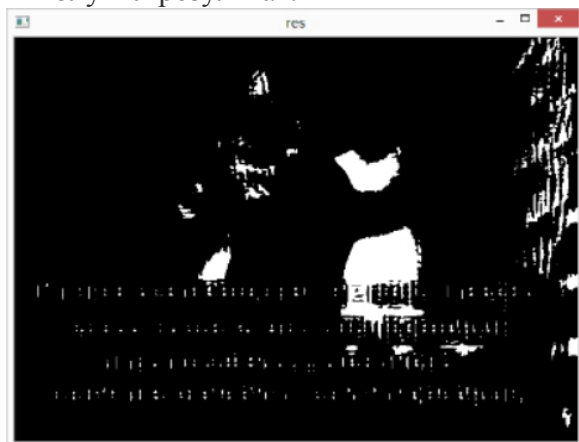


Рисунок 1- результат через
`cvMatchTemplate()`

В зависимости от выбранного метода корреляции, местоположение шаблона можно вычислить по минимуму или максимуму значения на исходном изображении. Для этого используется функция `cvMinMaxLoc()`:

```
CVAPI(void) cvMinMaxLoc( constCvArr* arr,  
double* min_val, double* max_val, CvPoint*  
min_locCV_DEFAULT(NULL), CvPoint*  
max_locCV_DEFAULT(NULL), constCvArr*  
maskCV_DEFAULT(NULL) );  
где arr — массив (изображение) для поиска;  
min_val — указатель на переменную для  
сохранения минимального значения;  
max_val — указатель на переменную для  
сохранения максимального значения;  
min_loc — указатель на точку местоположения  
минимума;  
max_loc — указатель на точку местоположения  
максимума;  
mask — маска для выбора подмассива.  
для приведения результата корреляции в  
«видимый» формат используется нормировка  
изображения при помощи функции cvNormalize():  
CVAPI(void) cvNormalize( const CvArr* src,  
CvArr* dst, double a CV_DEFAULT(1.), double b  
CV_DEFAULT(0.), int norm_type
```

```
CV_DEFAULT(CV_L2), const CvArr* mask  
CV_DEFAULT(NULL) );  
где arr1, arr2 —исходное изображение;  
norm_type — тип нормировки;  
mask — маска.
```

Наконец мы используем функцию `cvRectangle()` для выделения прямоугольником найденных по шаблону букв «а».

Результат работы программы представлен на рисунках 1 и 2. На рисунке 1 показано исходное изображение, на рисунке 2 показано изображение с найденными и выделенными в прямоугольники буквами.

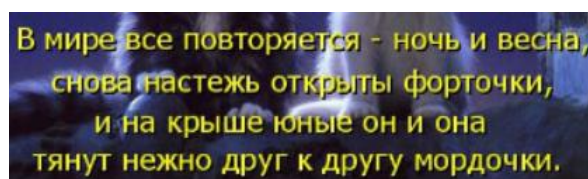


Рисунок 2 - Исходное изображение

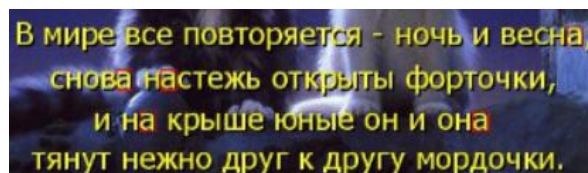


Рисунок 3 – Изображение, с найденными
буквами

Заключение

В процессе выполнения задания были оценены достоинства библиотеки компьютерного зрения OpenCV для решения задачи детектирования. В ходе выполнения работы было определено наличие заданной буквы на изображении, а также все найденные совпадения выделены координатами прямоугольников.

Список использованных источников

1. Цапко И.В., Власов А.В. Выделение объектов на изображениях методом поиска границ регионов // Автоматизация. Современные технологии. – 2015. № 9. С. 33-38.
2. Цапко И.В., Цапко С.Г., Гусакова Е.Г. Алгоритм создания трёхмерной сцены в 2DS MAX 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).
4. Gary Bradski, Adrian Kaehler. Learning OpenCV. Computer Vision with the OpenCV Library. - O'Reilly Media, 2008. – 580 p