Установка библиотеки OpenCV

Шокуров Антон В.
shokurov.anton.v@yandex.ru
http://машинноезрение.рф

28 сентября 2017 г.

Версия: 0.10

Аннотация

Рассмотрены различные способы установки бибилотеки OpenCV. Поддержка как Linux (Mac), так и Windows. Показано взаимодействие с QT.

1 Установка и настройка системы

Программы можно писать как на Си (Си++), так и на python. Материал данной заметки скорее ориентирован на разработку программ с использованием языков Си и Си++. Для написания программ на python нужно частично выполнить указания данной заметки, а потом установить соответствующий пакет для python.

Одной из целью является получения кросс платформенных программ. Сама библиотека OpenCV кросс платформена.

В первом подразделе рассмотрена система Linux, а во втором Windows.

1.1 Linux

В данном подразделе рассматривается операционная система Linux.

Может оказаться так, что у библиотека OpenCV уже установлена. Поэтому перед переходом в следующий параграф, проверь наличие библиотеки OpenCV в системе как это показано в одном из следующих параграфов.

Установка из репозитория Многие дистрибутивы Linux уже имеют пакет с OpenCV. Поэтому можно его установить от туда. Для этого нужна указать системе управления пакетами, что нужно установить пакеты относящиеся к OpenCV.

Желательно все. Такие пакеты обычно имеет имя начинающиеся с libopencv. Например, в семействе Debian такими пакетами будут: libopencv-core..., libopencv-imgproc..., libopencv-highgui... и тому подобное. Где к вместо троеточия идет номер версии библиотеки. В частности, может начинаться с 2.4. Детали точного названия не важны.

Более того, ввиду того, что библиотека нам нужна не только для запуска программ написанных с применением OpenCV, но и для разработки собственных программ, понадобятся пакеты отвечающие за сборку.

Для сборки программ в случае если все было сделано правильно:

```
1 g++ prog.cpp -std=c++11 `pkg-config opencv --cflags \
2 > --libs ` -o prog
```

Символы \и > обозначает то, что это продолжение строки. На самом деле достаточно все указать на одной строке (без этих символов).

prog.cpp это входной файл с кодом программы, а prog это выходной исполняемый файл.

Сборка Иногда бывает, что все-таки собрать библиотеку. Для этого могут быть разные причины: от отсутствия нужной версии библиотеки в репозитории, до необходимости детального изучения самого кода библиотеки.

Делается достаточно просто.

Сначало нужно скачать исходный код библиотеки. Для этого существуют как минимум два варианта. Либо готовый архив со страничке официального сайта либо с глобального места хранения исходных кодов (github).

На странице официального сайта (`http://opencv.org/releases.html`) нужно выбрать нужную версию библиотеки и выбрать для скачивания ссылку sources. Будет скачен архив. Далее его нужно будет разархивировать на диске с достаточным свободным местом.

Другой вариант это скачать сайт непосредственно из

https://github.com/opencv/opencv

Такой подход позволяет скачать максимально актуальную версию библиотеки. Актуальонсть определяется не только наличием самых последних плюшек, но и наличием новых ошибок. Для скачивания из такого хранилища можно либо непосредственно скачать сам архив с исходным кодом (чего будет достаточно) либо выудить данные используя специальную программу git, которую необходимо установить в систему в случае её отсутствия. Для сачивания данных с git нужно выбрать место куда скачать, например, папка ~/Downloads/opencv/ и далее вызвать git clone `https://github.com/opencv/opencv.git` Данная команда выудит исходный код в подпапку opencv. В данном случае код будет скачан в папку: ~/Downloads/opencv/opencv Чтобы в дальнейшем не путаться (например, если

необходимо работать с разными версиями библиотеки opencv) можно переименовать папку например в ~/Downloads/opencv/opencv-3.3.0/

Далее как и в предыдущем пункте переходим к процессу самой сборки библиотеки.

Пусть была скачена библиотека и размещена в папку
~/Downloads/opencv/opencv-3.3.0

Далее следует решить в какой папке будет создаваться код. Данная папка является временной, на время создания бинарников. Расположение этой папки не так важно так как скопировать бинарники всегда можно будет потом. Более того, установка бинарников в системную папку делается отдельно и не зависит от положения этой временной папки. Обычно её создают как подпапку и называют build.

Таким образом нужной перейти в папку ~/Downloads/opencv/opencv-3.3.0, создать подпапку build и перейти в неё.

После этих шагов в нужно сконфигурировать сборку. Можно выбрать например с форматом каких картинок будет уметь работать библиотека, какое будет ускорение и тому подобное.

Конфигурацию можно выполнить в текстовом режиме, но это скучно. Лучше использовать оконный конфигуратор. Для этого нужно в консоли вызвать cmake-gui .. из папки build. Далее нужно нажать кнопку конфигурировать (Configure), согласившись (Finish) с параметрами по умолчанию. Будет показан список параметров (кирпичный цвет ни очем плохом не говорит, просто разработчики любят такую цветовую гамму) Список параметров большой. Обычно можно делать сборку с параметрами по умолчанию. Но следует обратить внимания на параметры связанные с оконной системой QT и примеры программ (BUILD_EXAMPLES, INSTALL_C_EXAMPLES, INSTALL_PYTHON_EXAMPLES). После выбора параметров следует заново выполнить конфигурирование (нажать кнопку). После этого могут появится новый возможности. Например,при включении with_qt после повторного конфигурирования появятся параметры связанные непосредственно с qt.

Потом следует нажать кнопку сформировать (Generate), которая и создаст необходимые Makefileы для сборки библиотеки.

Далее следует выйти из данной программы и в консоли вызвать make. После чего можно пойти перекусить, процесс сборки достаточно продолжительный. Для ускорения можно указать make чтобы он выполнял работу в параллель: make -j 8, например укажет на использование 8 потоков для сборки.

Для сборки и запуска программ этого будет достаточно. Так, при компиляции нужно будет указывать компилятору расположение файлов заголовков, при сборки расположение бинарных библиотек.

При запуске тоже необходимо будет указать положение библиотек, но уже ди-

намических.

1.2 Windows

Установка из бинарников Windows в отличие от Linux разрабатывается централизовано, т.е. понятно, что за него отвечает фирма Microsoft. Поэтому программы под Windows бинарно совместимы. Последнее означает, что нет необходимости собирать библиотеку OpenCV под каждую версию Windows. Последнее влечет, что её достаточно собрать единожды и она будет работать "везде". Именно поэтому в случае Windows можно скачать нужные архив с официального сайта библиотеки: `http://opencv.org/releases.html` По крайней мере будет указана необходимая ссылка.

Этот архив скачивается и разархивируется в выбранном месте на диске. Программы установщика нету. поэтому нужно самостоятельно выбрать удобное расположение корневой папки библиотеки.

Самое сложное является настройка программы, которая используется для компиляции программ.

Для настройки проекта нужно указать пути к папка содержащем заголовочные файлы библиотеки. .. Тоже для библиотечных. Есть два типа бинарников: для отладки и для "выпускные". Их не следует путать, иначе программа может выдавать странные ошибки... и будет создаваться впечатление, что не работает программа, хотя на самом деле не правильно настроен компилятор/сборщик.

1.3 Программы

Запуск простейшей программы.

```
1  #include <opencv2/core/core.hpp>
2  #include <opencv2/highgui/highgui.hpp>
3
4  #include <iostream>
5
6  int main( int argc, char** argv )
7  {
8    if( argc < 3 )
9    {
10     std::cout <<"Need input and output "
11       "image file names!" << std::endl;
12     return -1;
13   }
14
```

```
15    cv::Mat I = cv::imread( argv[1],
16      cv::IMREAD_GRAYSCALE );
17    cv::imwrite( argv[2], I );
18    return 0;
19  }
```

Строчки в кавычках можно так не разбивать! Это сделано для размещения текста на странице!

Курс рассчитан на текущую версию библиотеки. Поэтому, если будут ошибки относительно cv::IMREAD_GRAYSCALE, то замени его на

CV_LOAD_IMAGE_GRAYSCALE

Запуск с окном из OpenCV

```
1  #include <opencv2/core/core.hpp>
2  #include <opencv2/highgui/highgui.hpp>
3
4  #include <iostream>
5
6  int main( int argc, char** argv )
7  {
8    if( argc < 2 )
9    {
10     std::cout <<"Need an image file name "
11       "to show!" << std::endl;
12     return -1;
13   }
14
15   cv::Mat I = cv::imread( argv[1], cv::IMREAD_COLOR );
16   cv::namedWindow( "My Window", cv::WINDOW_AUTOSIZE );
17   cv::imshow( "My Window", I );
18   cv::waitKey( 0 );
19   return 0;
20 }
```

Кросс-платформенность есть, но оконная система очень примитивная.

Запуск с QT.

Полноценная оконная система и кросс-платформенность.