

## Anleitung zur Einrichtung von *OpenCV*

Dieses Dokument beschreibt die Einrichtung und Funktionsüberprüfung von *OpenCV* im Kontext der Lehrveranstaltung *Medieninformatik LB*.

### Einrichtung

Dieser Abschnitt beschreibt die Einrichtung von *OpenCV* unter Ubuntu Linux.

#### Anmerkung zu möglicherweise bereits installierten Paketen

Bitte beachten Sie, dass die meisten über herkömmliche Paketverwaltungssoftware erhältlichen *OpenCV*-Pakete, z.B. `libopencv-*` über *apt*, zwar den Großteil der im Rahmen des Laboratoriums benötigten *OpenCV*-Funktionalität enthalten, zumeist aber nicht das für das Laboratorium essenzielle *nonfree*-Modul. Wenn Sie sicherstellen können, dass *OpenCV* mitsamt diesem Modul auf Ihrem System installiert ist, können Sie den nachfolgenden Installationsabschnitt überspringen (falls nicht, lesen Sie bitte unbedingt im nächsten Absatz weiter). Stellen Sie in diesem Fall lediglich sicher, dass die auf Ihrem System installierte *OpenCV*-Version mindestens 2.4.9 ist. Die Versionsnummer können Sie u.a. mit dem Aufruf

```
pkg-config --modversion opencv
```

überprüfen.

Wenn Sie nicht sicherstellen können, dass die auf Ihrem System installierte *OpenCV*-Version zur Nutzung im Rahmen des Laboratoriums geeignet ist (oder Sie sich dessen nicht sicher sind), ziehen Sie eine Deinstallation aller *OpenCV*-Pakete auf Ihrem System über die entsprechende Paketverwaltungssoftware in Betracht, sofern keine anderen Anwendungen diese Pakete benötigen. Parallelinstallationen im System führen oft zu Konflikten und schwer reproduzierbaren Problemen, weswegen von ersteren abzuraten ist.

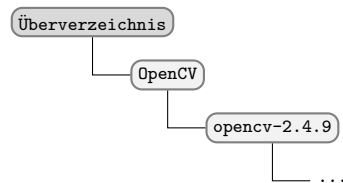
### Konfiguration

Es wird davon ausgegangen, dass auf dem System *gcc*, *g++* und *cmake* (aus dem Paket `build-essentials`) sowie `libgtk2.0-dev`, `libavcodec-dev`, `libavformat-dev`, `libswscale-dev`, `libjpeg-dev` und `libpng-dev` installiert sind.

Zuerst muss der Quelltext von *OpenCV* heruntergeladen werden. Dies kann u.a. durch Herunterladen und Entpacken der Sourcecode-ZIP-Datei der aktuellen Version von (Stand: 27.7.2014) <http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.9/opencv-2.4.9.zip/download> geschehen. Es wird empfohlen, die nachfolgende Ordnerstruktur in einem von Ihnen gewählten Überverzeichnis<sup>1</sup> zu verwenden:

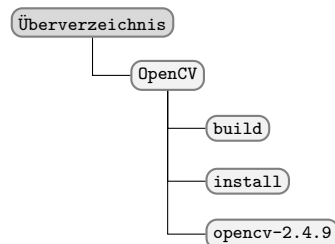
---

<sup>1</sup>Bitte geben Sie diesem Verzeichnis einen anderen Namen als `Überverzeichnis` – vermeiden Sie in jedem Fall Leer- und Sonderzeichen (insbesondere Umlaute) im **gesamten** Dateipfad des Überverzeichnisses



Der Inhalt des Ordners `opencv-2.4.9` entspricht dabei jenem des gleichnamigen Ordners aus der ZIP-Datei in entpackter Form.

Erstellen Sie für den Build-Prozess einen Ordner `build` und für die im nächsten Abschnitt beschriebene Installation einen Ordner `install`, falls eine lokale Installation gewünscht wird. Es wird empfohlen, dass diese beiden Ordner in der selben Hierarchieebene der Ordnerstruktur wie der Ordner `opencv-2.4.9` angelegt werden. Die Ordnerstruktur sieht dann wie folgt aus:



Wechseln Sie anschließend in das erstellte Verzeichnis `build` und rufen Sie `cmake` mit dem (bevorzugt relativen) Pfad des `opencv-2.4.9`-Ordners zur Konfiguration von *OpenCV* auf:

```
cmake ../opencv-2.4.9
```

Es ist notwendig, mittels `-D CMAKE_INSTALL_PREFIX=<Pfad>` zusätzlich einen Installationspfad anzugeben, falls eine lokale Installation gewünscht wird. Der obige Aufruf ist in diesem Fall durch den folgenden zu ersetzen:

```
cmake -D CMAKE_INSTALL_PREFIX=../install ../opencv-2.4.9
```

Zur Minimierung der Kompilierungszeit wird empfohlen, alle nicht benötigten Komponenten zu deaktivieren, z.B. über (**ohne** Zeilenumbrüche)

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D WITH_1394=OFF -D WITH_CUDA=OFF
-D WITH_CUFFT=OFF -D WITH_EIGEN=OFF -D WITH_GSTREAMER=OFF -D
WITH_JASPER=OFF -D WITH_OPENEXR=OFF -D WITH_PVAPI=OFF -D
WITH_GIGEAPI=OFF -D WITH_V4L=OFF -D WITH_LIBV4L=OFF -D
WITH_OPENCCL=OFF -D WITH_OPENCCLAMDFFT=OFF -D WITH_OPENCCLAMDBLAS=OFF
-D BUILD_SHARED_LIBS=OFF -D BUILD_opencv_apps=OFF -D
BUILD_PERF_TESTS=OFF -D BUILD_TESTS=OFF -D BUILD_opencv_gpu=OFF -D
BUILD_opencv_legacy=OFF -D BUILD_opencv_python=OFF -D
BUILD_opencv_superres=OFF -D CMAKE_INSTALL_PREFIX=../install
../opencv-2.4.9
```

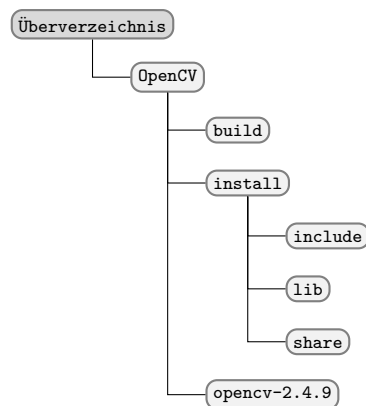
Falls Sie weitere Änderungen vornehmen, stellen Sie sicher, dass Sie keine für das Laboratorium relevanten Module über `-D BUILD_opencv_<Modulname>=OFF` deaktivieren.

## Installation

*OpenCV* kann unter Verwendung des im vorherigen Abschnitt konfigurierten `build`-Ordners mittels

```
make && make install
```

kompiliert und installiert werden. Der Ordner `install` enthält im Falle einer lokalen Installation die Header und Bibliotheken mit der folgenden Ordnerstruktur:



Beachten Sie, dass im Falle einer systemweiten Installation unter Umständen Super-User-Rechte für die Installation erforderlich sind.

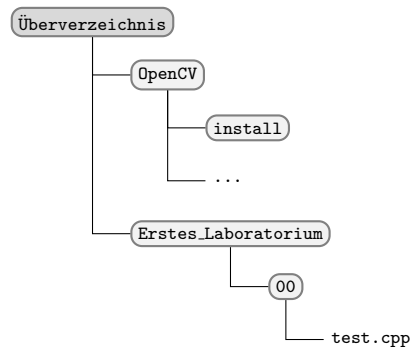
Bei Platzmangel können sowohl der `build`- als auch der `opencv-2.4.9`-Ordner gelöscht werden. Beachten Sie allerdings, dass dadurch eine Änderung der Installation (z.B. Aktivierung oder Deaktivierung von Modulen oder Funktionen) nicht mehr möglich ist und eine erneute Durcharbeitung dieses Abschnittes notwendig ist.

## Funktionsüberprüfung

Dieser Abschnitt beschreibt, wie die Funktionalität einer wie im vorigen Abschnitt beschrieben eingerichteten *OpenCV*-Installation überprüft werden kann.

### Minimal-Testprogramm

Legen Sie eine neue Datei `test.cpp` an. Es wird empfohlen, dabei die bestehende Ordnerstruktur aus dem vorherigen Abschnitt zu verwenden und derart zu erweitern, dass für das erste Laboratorium ein Unterordner und in diesem wiederum für das erste Beispiel ein weiterer Unterordner angelegt wird:



Achten Sie darauf, dass in keinem der Dateinamen und -pfade Leer- oder Sonderzeichen (insbesondere Umlaute) enthalten sind<sup>2</sup>.

Befüllen Sie anschließend `test.cpp` mit folgendem Inhalt:

```

#include <iostream>
#include "opencv2/core/core.hpp"

using namespace std;
using namespace cv;

int main(const int argc, const char * const argv[])
{
    cout << getBuildInformation() << endl;
    return 0;
}
  
```

Die Kernbestandteile (Klassen und Funktionen) von *OpenCV* werden dabei über `core.hpp` eingebunden; `cv::getBuildInformation` ist ein Beispiel für einen dieser Bestandteile – eine Hilfsfunktion, die Build-Informationen zurückliefert, die über `std::cout` auf der Standardausgabe, gefolgt von einem Zeilenumbruch (`std::endl`), ausgegeben werden.

### Makefile

Die im vorigen Abschnitt erstellte Datei `test.cpp` muss nun kompiliert und gelinkt werden. Bei Verwendung der Ordnerstruktur aus dem vorigen Abschnitt wird dabei folgendes Makefile als Muster empfohlen:

```

CC=g++

CFLAGS=-c -Wall `export
    ↪ PKG_CONFIG_PATH=../../OpenCV/install/lib/pkgconfig
    ↪ && pkg-config opencv --cflags `
  
```

<sup>2</sup>Zwar können nahezu alle Programme mit Leer- und Sonderzeichen umgehen, doch das im nachfolgenden Abschnitt verwendete *pkg-config* stellt in diesem Kontext eine Ausnahme dar

```
LDFLAGS=`export
↳ PKG_CONFIG_PATH=../../OpenCV/install/lib/pkgconfig
↳ && pkg-config opencv --libs`

all: EL00a

EL00a: test
_____$(CC) test.o -o EL00a.exe $(LDFLAGS)

test: test.cpp
_____$(CC) $(CFLAGS) test.cpp

clean:
_____rm -f *.o *.exe
```

Beachten Sie die Tabulatoren innerhalb der Targets sowie die Zeilenumbrüche! Als C++-Compiler wird `g++` (gesetzt über die Variable `CC`) verwendet. Die entsprechenden Kommandozeilenparameter zur Einbindung der *OpenCV*-Pfade und -Bibliotheken in der richtigen Reihenfolge werden hier durch *pkg-config* generiert, das auf das bei der Installation erzeugte Verzeichnis mit Konfigurationsinformationen zu *OpenCV* zurückgreift.

Durch den Aufruf von

`make`

im Verzeichnis von `test.cpp` wird nun das Programm `EL00a.exe` erzeugt.

## Überprüfung

Durch Ausführen des im vorigen Abschnitt erzeugten Beispielprogrammes über `./EL00a.exe`

kann überprüft werden, ob die *OpenCV*-Installation funktionsfähig (und das Beispielprogramm samt Makefile korrekt) ist – werden ohne weitere Fehler die Build-Informationen ähnlich dem unten gezeigten Beispiel ausgegeben, war die Installation erfolgreich und das Beispielprogramm funktioniert.

```
General configuration for OpenCV 2.4.9 =====
Version control:                unknown

Platform:
Host:                            Linux 3.13.0-32-generic x86_64
CMake:                            2.8.12.2
CMake generator:                 Unix Makefiles
CMake build tool:                /usr/bin/make
Configuration:                   RELEASE

[..]

Install path:                    /home/aunterwe/MI/OpenCV/install
cvconfig.h is in:                /home/aunterwe/MI/OpenCV/build
-----
```