OPENCV

OpenCV is big community library (set of already written programs) for Computer Vision applications, focusing on processing of images and videos. One should find it quite useful. It is possible to use a few built-in CUDA functions, or your own device kernels (firstly, please refer to CUDA repository and 00 step-by-step introduction, for setting up nvcc environment).

1) go to OpenCV install website "https://docs.opencv.org/master/d7/d9f/tutorial_linux_install.html", and follow instructions:

```
#sudo aptitude install build-essential
#sudo aptitude install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
#sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev
libjasper-dev libdc1394-22-dev
#sudo aptitude install cmake-qt-gui doxygen liblapack3  liblapack-dev libfftw3-dev
#sudo aptitude install libblas3 libblas-dev libopenblas-dev
```

2) clone project:

```
#git clone https://github.com/opencv/opencv.git
#cd opencv/ && mkdir build && cd build && sudo cmake-gui
```

set your sources path (...../opencv), and where build configuration (...../opencv/build); check CheckBox "Grouped", and "Advanced". For building GPU-accelerated functions just point NVCC-compatible C++ compiler:

```
CMAKE->CMAKE_CXX_COMPILER /usr/bin/g++-4.9
CMAKE->CMAKE_C_COMPILER /usr/bin/gcc-4.9
```

some useful functionalities:

```
ENABLE->ENABLE_CXX11
ENABLE->ENABLE_FAST_MATH
BUILD->BUILD_JPEG
BUILD->BUILD_DOC
BUILD->BUILD_EXAMPLES
INSTALL->INSTALL_C_EXAMPLES
```

3) click Button "Configure", check for errors (often there is some lack of libraries), "Generate", and close

```
#make -j`nproc`
```

4) compile and run basic tutorial from: "https://docs.opencv.org/master/db/df5/tutorial_linux_gcc_cmake.html"

5) optionally one can compile and install additional modules:

```
#git clone https://github.com/opencv/opencv_contrib.git
```