OpenCV 3.1 cross compilation on Linux A15

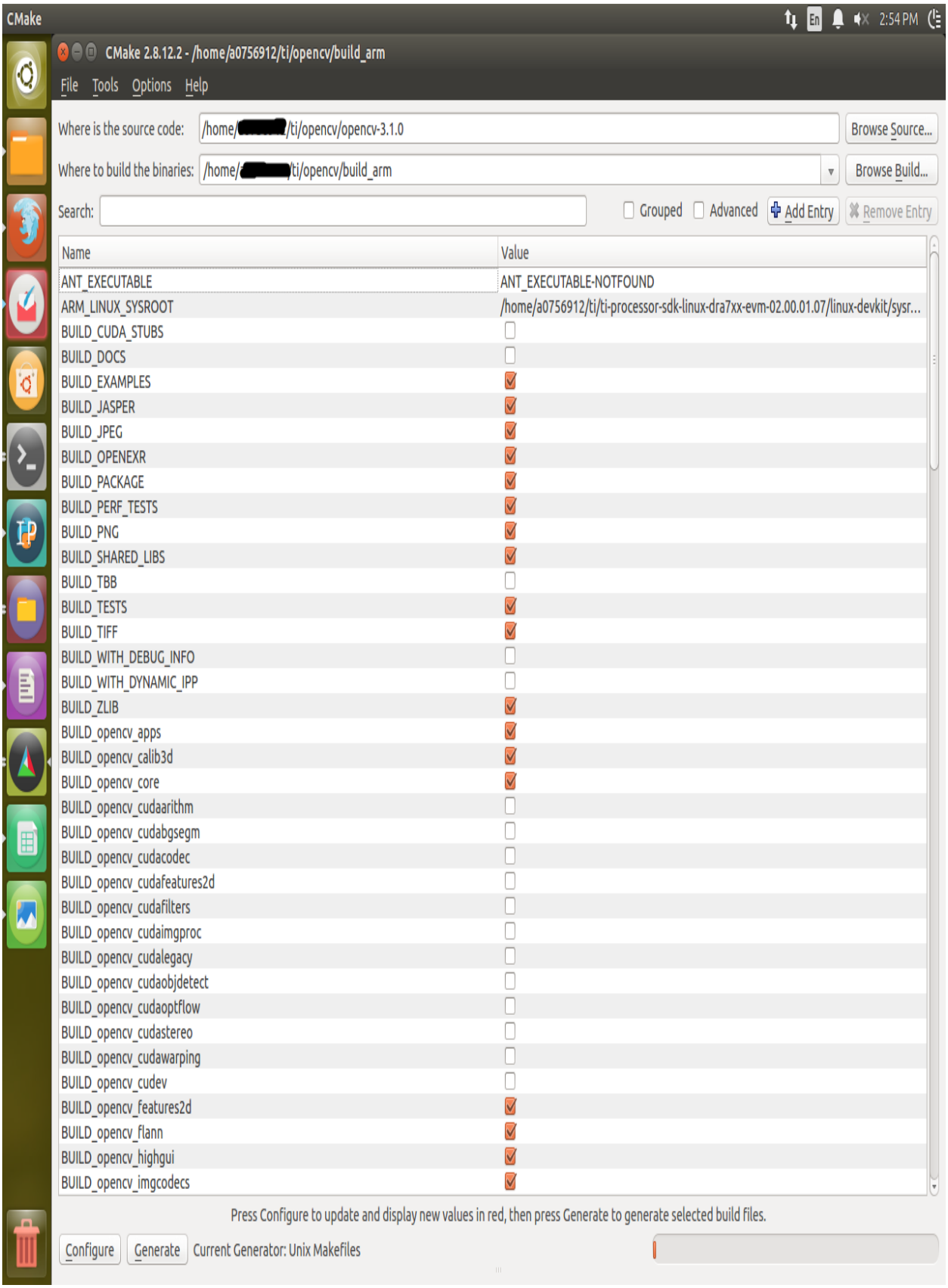
Instructions:

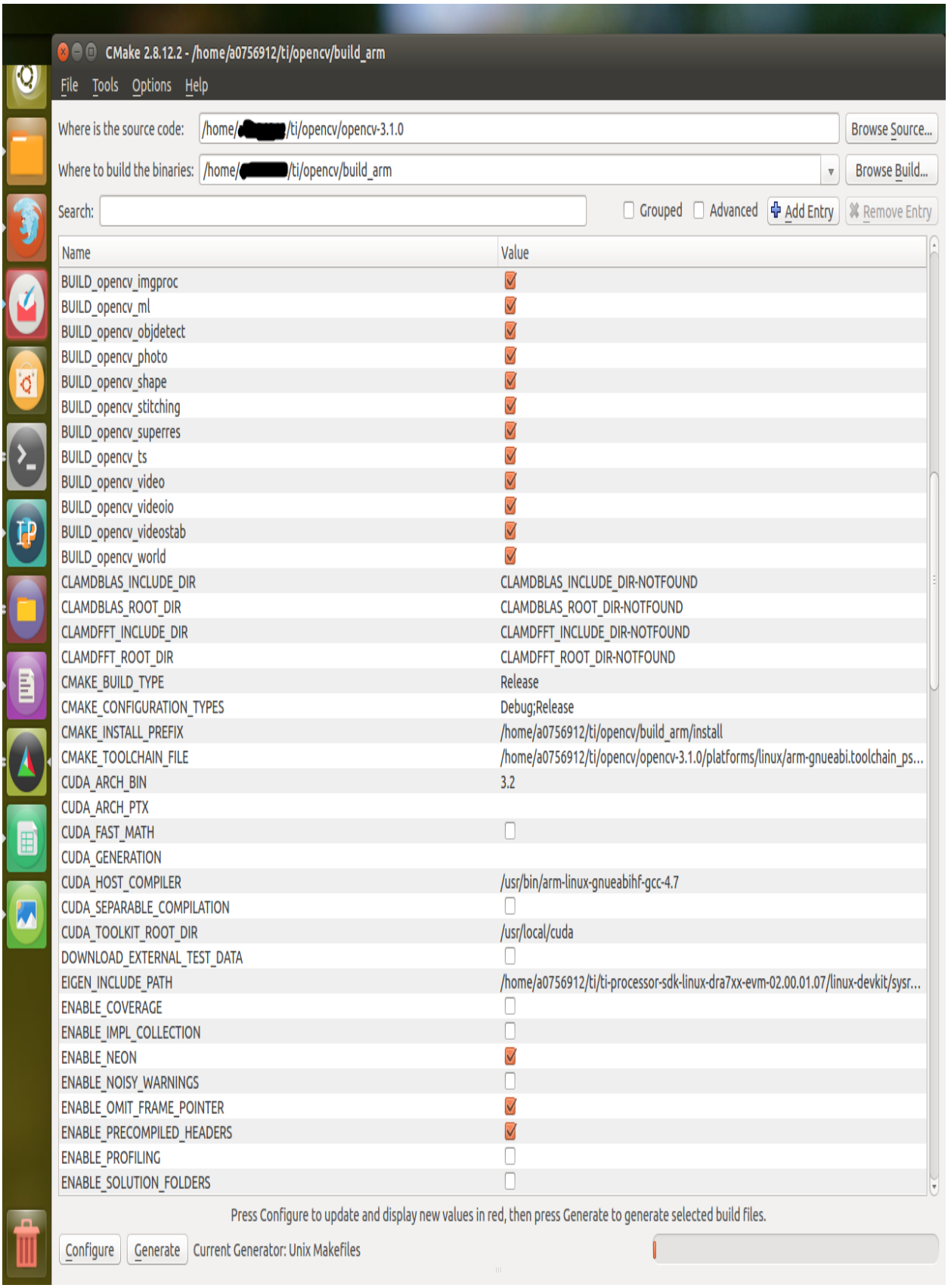
1. **Required host machine:** Ubuntu 10.04 or 12.04 or 14.04

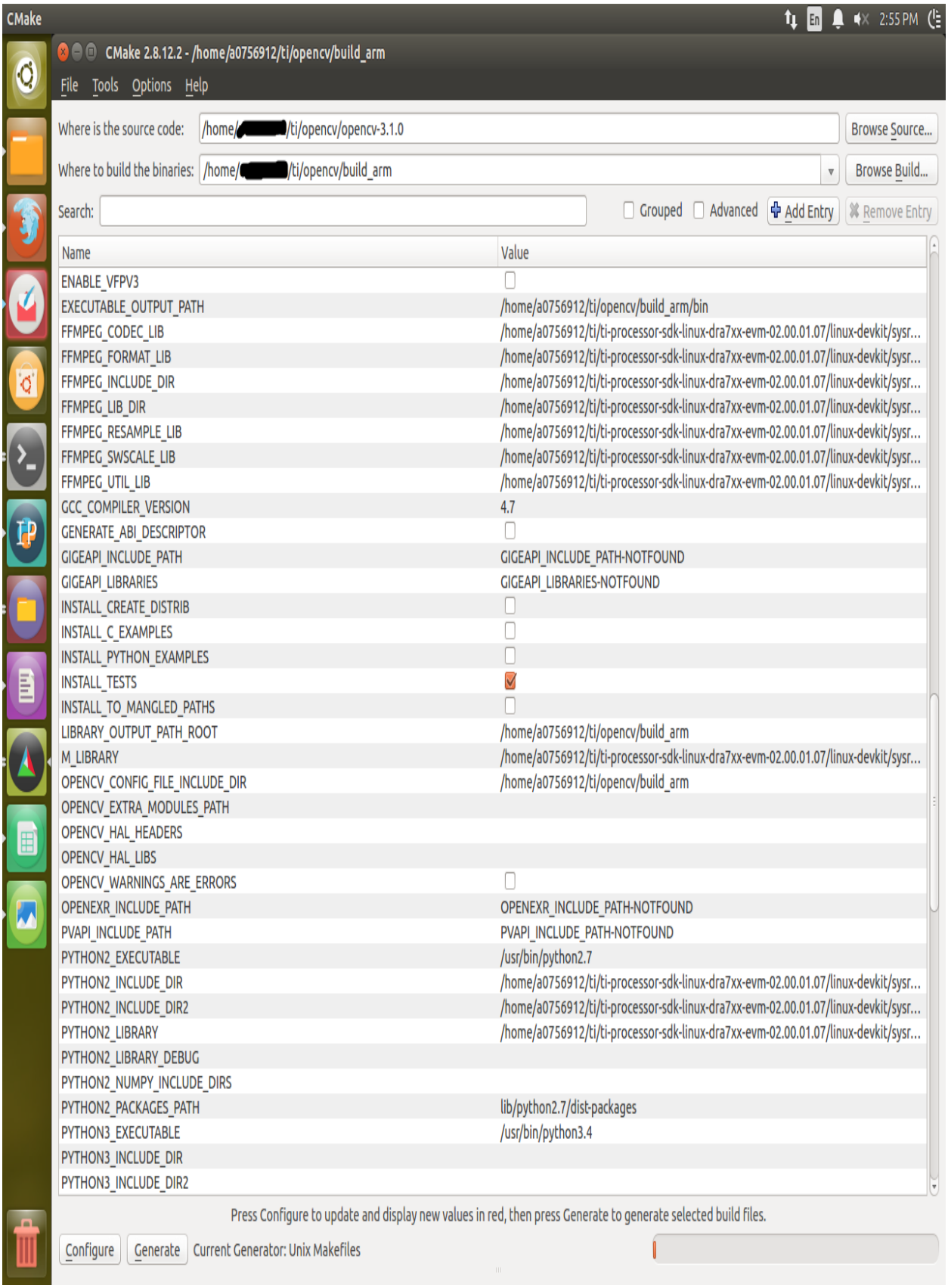
   Note: to copy from terminal – shift + ctrl + c and to paste into terminal – shift + ctrl + v
2. Let's make a folder called 'ti' in the home folder and keep the OpenCV package and Vision SDK package there
   a. cd ~
   b. mkdir ti
3. Download and install Vision SDK 02.11
4. Please follow the steps mentioned in *VisionSDK_LinuxUserGuide.pdf* in <vision_sdk_install_path>/VISION_SDK_02_08_00_00/vision_sdk/linux/docs folder
5. It is expected that the user has installed the A15 compiler as mentioned in section 2.2.1 in the user guide
6. So, the linaro toolchain should be found inside ~/ti/VISION_SDK_02_08_00_00/ti_components/os_tools/linux/linaro
7. **OpenCV setup + applying patch**
   a. cd ~/ti
   b. mkdir opencv
   c. cd opencv
   d. clone tiopencv repository using
      i. git clone https://github.com/opencv/opencv.git
   e. Go to the tag ticv3.1_00.04.02.00
   f. Create a build directory called 'build_arm'
8. **Arm-toolchain file**
   a. The Linux ARM cmake toolchain is part of the patch.
   b. It should be found inside the <opencv_path>/platforms/generic/ named 'arm-gnueabi.toolchain_vsdk.cmake'
   c. The build depends on the following components:
      i. GCC
      ii. OpenCL (part of VSDK Linux targetfs)
      iii. VXLIB
      iv. DSP CGT 8.1.0
   d. Go through the toolchain and edit the appropriate paths to point to the appropriate modules path mentioned earlier

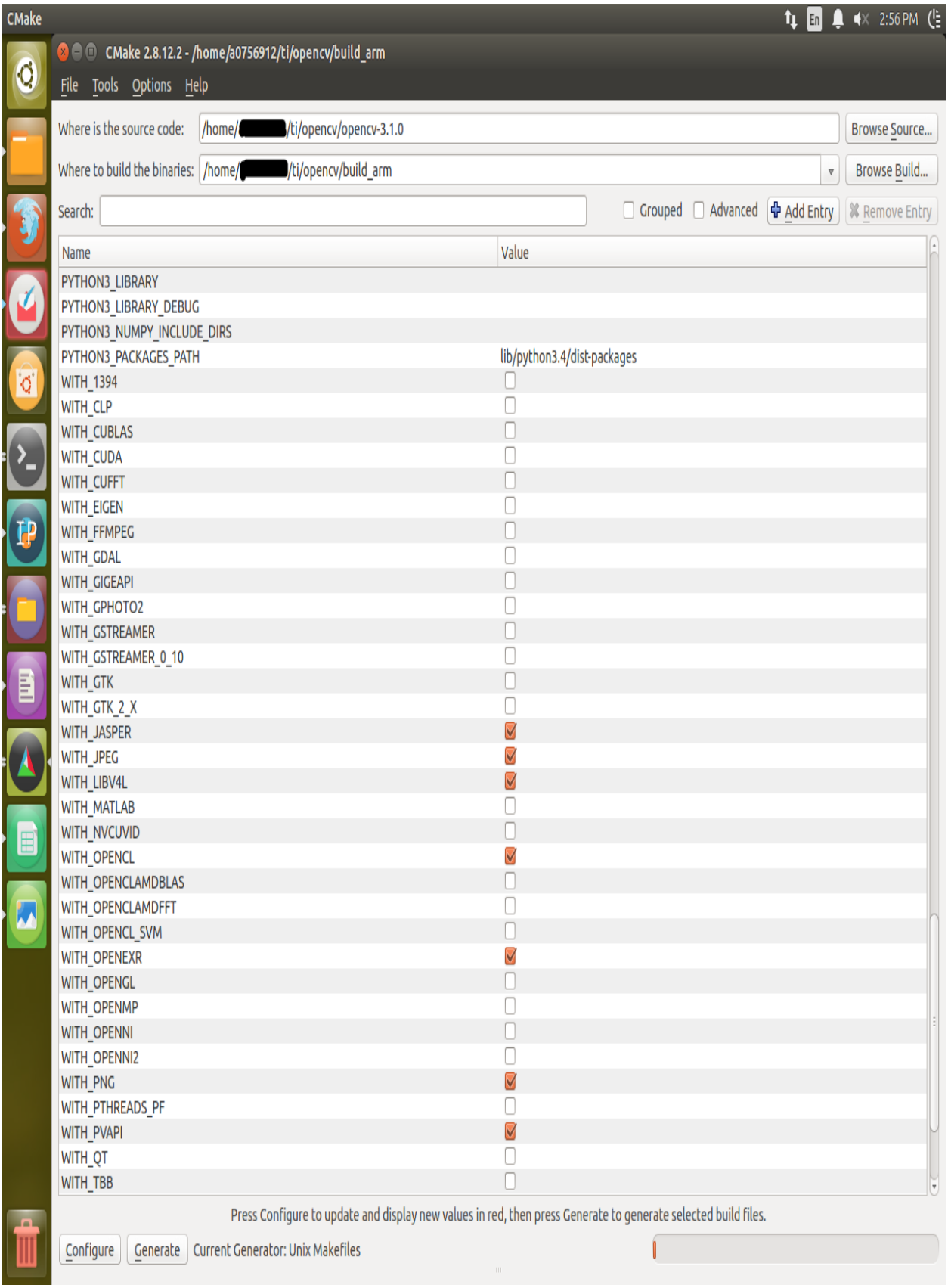
   Note: Let's keep the FLOAT_ABI_SUFFIX as 'hf' only and not disturb it unless required

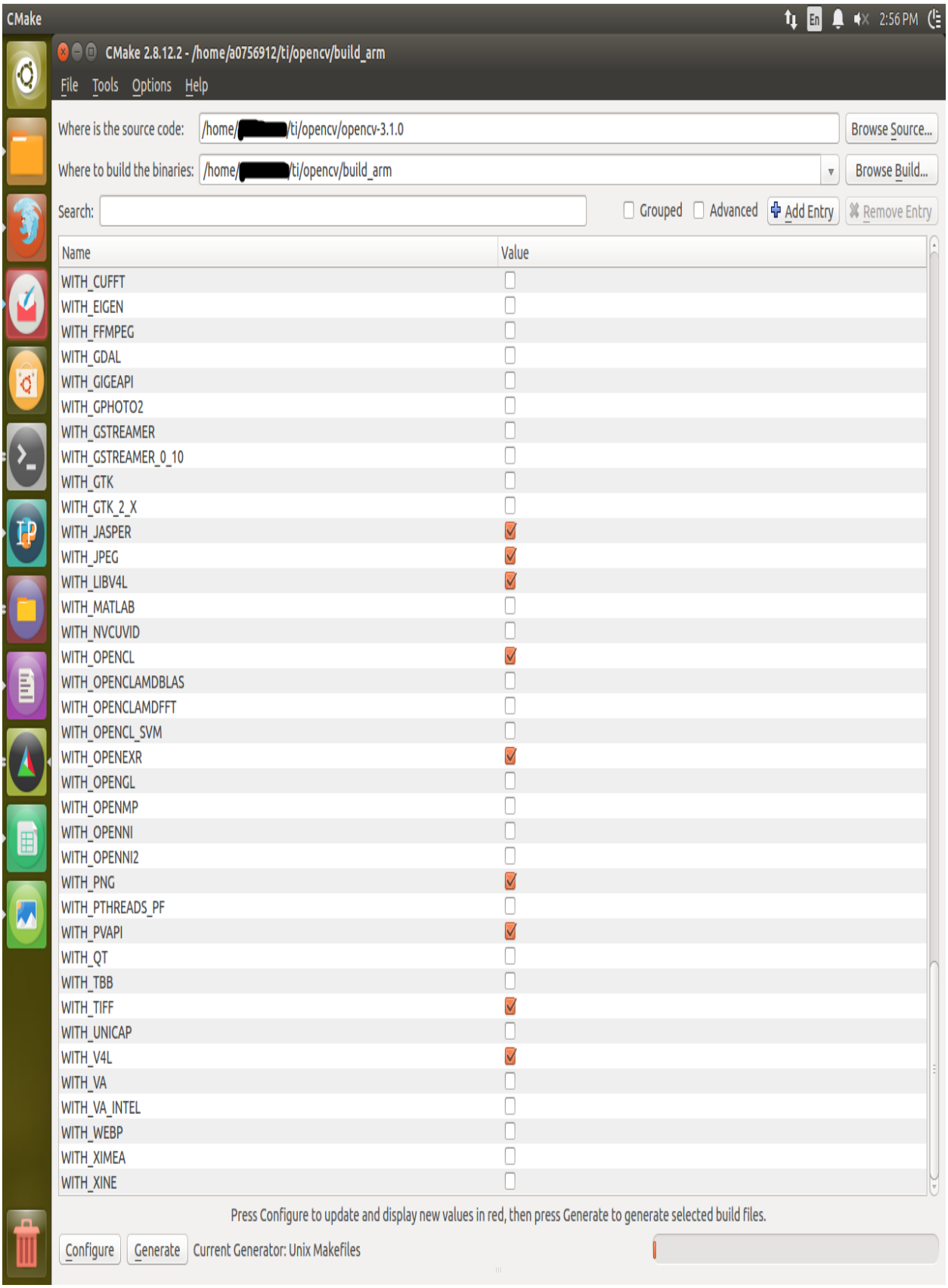
9. Install 32-bit libraries if the host machine is 64-bit and the downloaded toolchain is 32-bit
    a. **Cmd:** sudo apt-get install ia32-libs
10. **Download cmake**
    a. Open Terminal – ctrl + alt + t
    b. **Cmd:** sudo apt-get install cmake
    c. **Cmd:** sudo apt-get install cmake-qt-gui
11. **Configuring OpenCV project**
    a. Open cmake gui
        i. **Cmd:** cmake-qt-gui
    b. **Choose source path**: /home/<username>/ti/opencv/tiopencv
    c. **Choose build path**: /home/<username>/ti/opencv/build_arm
    d. Press configure -> select toolchain file -> location: ~/ti/opencv/tiopencv/platforms/linux/arm-gnueabi.toolchain_vsdk.cmake (~ stands for /home/<username>)
    e. Select reqd. modules as follows:


CMake
2:54 PM
CMake 2.8.12.2 - /home/a0756912/ti/opencv/build_arm
File Tools Options Help
Where is the source code: /home/[redacted]/ti/opencv/opencv-3.1.0
Browse Source...
Where to build the binaries: /home/[redacted]/ti/opencv/build_arm
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Name
Value
ANT_EXECUTABLE
ANT_EXECUTABLE-NOTFOUND
ARM_LINUX_SYSROOT
/home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
BUILD_CUDA_STUBS
BUILD_DOCS
BUILD_EXAMPLES
BUILD_JASPER
BUILD_JPEG
BUILD_OPENEXR
BUILD_PACKAGE
BUILD_PERF_TESTS
BUILD_PNG
BUILD_SHARED_LIBS
BUILD_TBB
BUILD_TESTS
BUILD_TIFF
BUILD_WITH_DEBUG_INFO
BUILD_WITH_DYNAMIC_IPP
BUILD_ZLIB
BUILD_opencv_apps
BUILD_opencv_calib3d
BUILD_opencv_core
BUILD_opencv_cudaarithm
BUILD_opencv_cudabgsegm
BUILD_opencv_cudacodec
BUILD_opencv_cudafeatures2d
BUILD_opencv_cudafilters
BUILD_opencv_cudaimgproc
BUILD_opencv_cudalegacy
BUILD_opencv_cudaobjdetect
BUILD_opencv_cudaoptflow
BUILD_opencv_cudastereo
BUILD_opencv_cudawarping
BUILD_opencv_cudev
BUILD_opencv_features2d
BUILD_opencv_flann
BUILD_opencv_highgui
BUILD_opencv_imgcodecs
Press Configure to update and display new values in red, then press Generate to generate selected build files.
Configure
Generate
Current Generator: Unix Makefiles


CMake 2.8.12.2 - /home/a0756912/ti/opencv/build_arm
File Tools Options Help
Where is the source code: /home/[redacted]/ti/opencv/opencv-3.1.0
Browse Source...
Where to build the binaries: /home/[redacted]/ti/opencv/build_arm
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Name
Value
BUILD_opencv_imgproc
BUILD_opencv_ml
BUILD_opencv_objdetect
BUILD_opencv_photo
BUILD_opencv_shape
BUILD_opencv_stitching
BUILD_opencv_superres
BUILD_opencv_ts
BUILD_opencv_video
BUILD_opencv_videoio
BUILD_opencv_videostab
BUILD_opencv_world
CLAMDBLAS_INCLUDE_DIR CLAMDBLAS_INCLUDE_DIR-NOTFOUND
CLAMDBLAS_ROOT_DIR CLAMDBLAS_ROOT_DIR-NOTFOUND
CLAMDFFT_INCLUDE_DIR CLAMDFFT_INCLUDE_DIR-NOTFOUND
CLAMDFFT_ROOT_DIR CLAMDFFT_ROOT_DIR-NOTFOUND
CMAKE_BUILD_TYPE Release
CMAKE_CONFIGURATION_TYPES Debug;Release
CMAKE_INSTALL_PREFIX /home/a0756912/ti/opencv/build_arm/install
CMAKE_TOOLCHAIN_FILE /home/a0756912/ti/opencv/opencv-3.1.0/platforms/linux/arm-gnueabi.toolchain_ps...
CUDA_ARCH_BIN 3.2
CUDA_ARCH_PTX
CUDA_FAST_MATH
CUDA_GENERATION
CUDA_HOST_COMPILER /usr/bin/arm-linux-gnueabihf-gcc-4.7
CUDA_SEPARABLE_COMPILATION
CUDA_TOOLKIT_ROOT_DIR /usr/local/cuda
DOWNLOAD_EXTERNAL_TEST_DATA
EIGEN_INCLUDE_PATH /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
ENABLE_COVERAGE
ENABLE_IMPL_COLLECTION
ENABLE_NEON
ENABLE_NOISY_WARNINGS
ENABLE_OMIT_FRAME_POINTER
ENABLE_PRECOMPILED_HEADERS
ENABLE_PROFILING
ENABLE_SOLUTION_FOLDERS
Press Configure to update and display new values in red, then press Generate to generate selected build files.
Configure
Generate
Current Generator: Unix Makefiles


CMake
2:55 PM
CMake 2.8.12.2 - /home/a0756912/ti/opencv/build_arm
File Tools Options Help
Where is the source code: /home/[redacted]/ti/opencv/opencv-3.1.0
Browse Source...
Where to build the binaries: /home/[redacted]/ti/opencv/build_arm
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Name
Value
ENABLE_VFPV3
EXECUTABLE_OUTPUT_PATH /home/a0756912/ti/opencv/build_arm/bin
FFMPEG_CODEC_LIB /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
FFMPEG_FORMAT_LIB /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
FFMPEG_INCLUDE_DIR /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
FFMPEG_LIB_DIR /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
FFMPEG_RESAMPLE_LIB /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
FFMPEG_SWSCALE_LIB /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
FFMPEG_UTIL_LIB /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
GCC_COMPILER_VERSION 4.7
GENERATE_ABI_DESCRIPTOR
GIGEAPI_INCLUDE_PATH GIGEAPI_INCLUDE_PATH-NOTFOUND
GIGEAPI_LIBRARIES GIGEAPI_LIBRARIES-NOTFOUND
INSTALL_CREATE_DISTRIB
INSTALL_C_EXAMPLES
INSTALL_PYTHON_EXAMPLES
INSTALL_TESTS
INSTALL_TO_MANGLED_PATHS
LIBRARY_OUTPUT_PATH_ROOT /home/a0756912/ti/opencv/build_arm
M_LIBRARY /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
OPENCV_CONFIG_FILE_INCLUDE_DIR /home/a0756912/ti/opencv/build_arm
OPENCV_EXTRA_MODULES_PATH
OPENCV_HAL_HEADERS
OPENCV_HAL_LIBS
OPENCV_WARNINGS_ARE_ERRORS
OPENEXR_INCLUDE_PATH OPENEXR_INCLUDE_PATH-NOTFOUND
PVAPI_INCLUDE_PATH PVAPI_INCLUDE_PATH-NOTFOUND
PYTHON2_EXECUTABLE /usr/bin/python2.7
PYTHON2_INCLUDE_DIR /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
PYTHON2_INCLUDE_DIR2 /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
PYTHON2_LIBRARY /home/a0756912/ti/ti-processor-sdk-linux-dra7xx-evm-02.00.01.07/linux-devkit/sysr...
PYTHON2_LIBRARY_DEBUG
PYTHON2_NUMPY_INCLUDE_DIRS
PYTHON2_PACKAGES_PATH lib/python2.7/dist-packages
PYTHON3_EXECUTABLE /usr/bin/python3.4
PYTHON3_INCLUDE_DIR
PYTHON3_INCLUDE_DIR2
Press Configure to update and display new values in red, then press Generate to generate selected build files.
Configure
Generate
Current Generator: Unix Makefiles


CMake
2:56 PM
CMake 2.8.12.2 - /home/a0756912/ti/opencv/build_arm
File Tools Options Help
Where is the source code: /home/[redacted]/ti/opencv/opencv-3.1.0
Browse Source...
Where to build the binaries: /home/[redacted]/ti/opencv/build_arm
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Name
Value
PYTHON3_LIBRARY
PYTHON3_LIBRARY_DEBUG
PYTHON3_NUMPY_INCLUDE_DIRS
PYTHON3_PACKAGES_PATH lib/python3.4/dist-packages
WITH_1394
WITH_CLP
WITH_CUBLAS
WITH_CUDA
WITH_CUFFT
WITH_EIGEN
WITH_FFMPEG
WITH_GDAL
WITH_GIGEAPI
WITH_GPHOTO2
WITH_GSTREAMER
WITH_GSTREAMER_0_10
WITH_GTK
WITH_GTK_2_X
WITH_JASPER
WITH_JPEG
WITH_LIBV4L
WITH_MATLAB
WITH_NVCUVID
WITH_OPENCL
WITH_OPENCLAMDBLAS
WITH_OPENCLAMDFFT
WITH_OPENCL_SVM
WITH_OPENEXR
WITH_OPENGL
WITH_OPENMP
WITH_OPENNI
WITH_OPENNI2
WITH_PNG
WITH_PTHREADS_PF
WITH_PVAPI
WITH_QT
WITH_TBB
Press Configure to update and display new values in red, then press Generate to generate selected build files.
Configure
Generate
Current Generator: Unix Makefiles


CMake
2:56 PM
CMake 2.8.12.2 - /home/a0756912/ti/opencv/build_arm
File Tools Options Help
Where is the source code: /home/[redacted]/ti/opencv/opencv-3.1.0
Browse Source...
Where to build the binaries: /home/[redacted]/ti/opencv/build_arm
Browse Build...
Search:
Grouped
Advanced
Add Entry
Remove Entry
Name
Value
WITH_CUFFT
WITH_EIGEN
WITH_FFMPEG
WITH_GDAL
WITH_GIGEAPI
WITH_GPHOTO2
WITH_GSTREAMER
WITH_GSTREAMER_0_10
WITH_GTK
WITH_GTK_2_X
WITH_JASPER
WITH_JPEG
WITH_LIBV4L
WITH_MATLAB
WITH_NVCUVID
WITH_OPENCL
WITH_OPENCLAMDBLAS
WITH_OPENCLAMDFFT
WITH_OPENCL_SVM
WITH_OPENEXR
WITH_OPENGL
WITH_OPENMP
WITH_OPENNI
WITH_OPENNI2
WITH_PNG
WITH_PTHREADS_PF
WITH_PVAPI
WITH_QT
WITH_TBB
WITH_TIFF
WITH_UNICAP
WITH_V4L
WITH_VA
WITH_VA_INTEL
WITH_WEBP
WITH_XIMEA
WITH_XINE
Press Configure to update and display new values in red, then press Generate to generate selected build files.
Configure
Generate
Current Generator: Unix Makefiles

f. OpenMP and pthread can be enabled if multi-core processing is desired

Note: all the opencv kernels may not be parallelized

g. Click configure
h. Click generate
i. Close cmake
j. **Cmd:** make –j4
k. **Cmd:** make install

12. If the above build fails, with an assembler statement saying "offset out of range", we need the 64-bit version of the same toolchain
    a. Get the toolchain package
        i. **Cmd:** sudo gedit /etc/apt/sources.list
        ii. Go to eof
        iii. Add the line – deb http://cz.archive.ubuntu.com/ubuntu trusty main universe
        iv. Save and close the file
        v. **Cmd:** sudo apt-get update
    b. **Cmd:** sudo apt-get install gcc-4.7-arm-linux-gnueabihf
    c. **Cmd:** sudo apt-get install g++-4.7-arm-linux-gnueabihf
13. Edit the arm-toolchain file if step 12 was followed
    a. Open terminal – ctrl + alt + t
    b. gedit ~/ti/opencv/opencv-3.1.0/platforms/linux/arm-gnueabi.toolchain.cmake
    c. Edit the following lines
        i. find_program(CMAKE_C_COMPILER NAMES arm-linux-gnueabi${FLOAT_ABI_SUFFIX}-gcc-4.7)
        ii. find_program(CMAKE_CXX_COMPILER NAMES arm-linux-gnueabi${FLOAT_ABI_SUFFIX}-g++-4.7)
        iii. set(ARM_LINUX_SYSROOT /usr/arm-linux-gnueabihf PATH "ARM cross compilation system root")
14. Go to project configuration
    a. Clear cache
    b. Repeat step 11(Configuring OpenCV project..)
    c. Now the build should be successful
15. Installing the libraries
    a. Run this command
        i. Cmd: make install
    b. This will install the headers and libraries in *install* folder inside the *build_arm* folder
    c. The demo or test executables can be found inside the *bin* folder of *build_arm* folder

List of supported modules

Below list of modules are supported for BIOS build

S. No	Modules
1	calib3d
2	core
3	features2d
4	flann
5	imgcodecs
6	imgproc
7	ml
8	objdetect
9	photo
10	shape
11	stitching
12	superres
13	video
14	videostab

List of OpenCV functions accelerated using OpenCL on DSP

For performance data refer to the performance excel sheets:

1. vayu_arm_linux_opencv_test_report.xls – OpenCV tests for arm linux
2. vayu_arm_bios_opencv_test_report.xls – OpenCV tests for arm linux
3. OpenCV_offload_DSP_profiling.xlsx – OpenCV tests for DSP accelerated functions

Function	Constraints	Introduced in VSDK version
cv::erode	8-bit single channel input; 8-bit single channel output; only 3x3 structuring is supported	2.11
cv::dilate	8-bit single channel input; 8-bit single channel output; only 3x3 structuring is supported	2.11
cv::GaussianBlur	8-bit single channel input; 8-bit single channel output; only 3x3	2.11

	structuring is supported	
cv::medianBlur	8-bit single channel input; 8-bit single channel output; only 3x3 structuring is supported	2.11
cv::LUT	8-bit single channel input	2.12
cv::MorphologyEx	8-bit single channel input; single iteration only; default anchor only supported	2.12
cv::PyrDown	8-bit single channel input; difference in implementation compared to opencv – the difference is in Gaussian kernel which is 1/256 of | 1 4 6 4 1| | 4 16 24 16 4| | 6 24 36 24 6| | 4 16 24 16 4| | 1 4 6 4 1|	2.12
cv::Sobel	8-bit single channel input; only 3x3 structuring is supported; sobel xy calculates magnitude also and output is 16-bit single channel	2.12
cv::Resize	8-bit single channel input; only bilinear and downscaling to half size is supported	2.12
cv::Integral	8-bit single channel input;	2.12
cv::CalcHist	8-bit single channel input; only 256 bins and range of 256 is supported	2.12
cv::EqualizeHist	8-bit single channel input	2.12

Accelerating more VLIB/VXLIB kernels

1. Before Building new OpenCL wrappers for optimized DSP kernels

- In order to build OpenCL
  - The OpenCL cross compiler, 'clocl' is required (this can be found inside the ti_components directory of Vision SDK) and
  - The appropriate DSP symbols
- The DSP symbols directory needs to be exported using an environment variable, 'TARGET_ROOTDIR'
- Please follow the FeatureSpecificguidelines on OpenCX for setting up the 'TARGET_ROOTDIR'
- Also, point to the CLOCL path in the OpenCV Linux cmake toolchain

2. Building new OpenCL wrappers for optimized DSP kernels

- The above list of OpenCV functions was accelerated using TI VXLIB C66X kernels.
- The OpenCL wrapper around these optimized DSP kernels could be found inside each opencv module inside 'src/ti_opencl' folder

    - i.e
        - <path to tiopencv>/modules/imgproc/src/ti_opencl
- OpenCL 1.1 allows calling any C function from a target library
    - In order to do this, an interface to the function is needed and
    - The target library to link with
- If new OpenCL wrappers for other VXLIB kernels are going to be added to any OpenCV module
    - Create appropriate .cl file
    - Edit the interface file to add prototype of the function to be called
    - Then do 'cmake' and make
    - The OpenCV build system is improved to automatically build all opencl files inside the 'ti_opencl' directory
- In order to link with new DSP libraries such as VLIB
    - Add appropriate interface files in ti_opencl directory
    - Edit 'OpenCVModule.cmake' inside <path to tiopencv>/cmake
        - Search for the string 'CLOCL_CMD
        - It could be seen that a command is formed to build the OpenCL file
        - Add the new library path in the command to link with it

**Note: In order to build the TI OpenCL files, pleae run 'cmake' which builds it.*

3. Utilizing the OpenCL wrappers

- After the TI OpenCL files are build the binary is stored in a character array in a correspding header inside the <opencv_build_dir_path>/modules/<module_name>/ti_opencl/<header>
- Include the header in the appropriate source file.
    - E.g. See morph.cpp
    - Add an equivalent function to call TI OpenCL file e.g. 'ticl_morph'
    - Acquire the OpenCL device using 'ocl::Device::getDefault'
    - Create a 'ocl::ProgramBinary' object from the binary string
    - Set appropriate global and local sizes
    - Create a 'ocl::Kernel' object from 'ocl::ProgramBinary ' object
    - Set the kernel arguments
    - Finally use the method 'run' of 'ocl::Kernel' object to offload and run the kernel
- Make sure that the 'ticl_<kernel_name>' function is called inside the appropriate OpenCV C++ function.