Android and OpenCV Tutorial

Computer Vision Lab Tutorial
26 September 2013

Lorenz Meier, Amaël Delaunoy, Kalin Kolev
Tutorial Content

- Strengths / Weaknesses of Android
- Java vs Native
- Overview of available SDKs
- Complete example
- Assignment
- Project plan coaching
- Question session
Android Software Development Kits

Software Development Kit (SDK)
Native Development Kit (NDK)
Android

- Android 4.3 Jelly Bean
- Targeted compatibility version: 4.1
Java SDK

- Easy to setup and get started
- Typically only mean of accessing the camera
- Slow for core vision algorithms
- Slow for some sensor access functions
Android NDK (C/C++, GNU/Linux API)

• Still easy enough to get started
• Offers full OpenCV functionality on the device
• Important: Have NDKROOT set as environment variable in Eclipse

```
export PATH=/usr/local/bin:/usr/local/sbin:/Users/lomeier/android/android-ndk-r9:$PATH
export NDKROOT=/Users/lomeier/android/android-ndk-r9
chmod u+x /Users/lomeier/android/android-ndk-r9/ndk-build.cmd
```

Computer Vision Toolkits for Android

OpenCV
Qualcomm Vuforia
Qualcomm FastCV
OpenVX
OpenCV on Android

• Main reference
  http://opencv.org/platforms/android.html

• Watch out! API has changed, use only documentation and tutorials from 2012/2013

• Start off with the OpenCV Tutorial #3: Native OpenCV

http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/android_dev_intro.html
Qualcomm Vuforia SDK

• Virtual Reality SDK for any Android / iOS device
  https://developer.qualcomm.com/download

http://www.youtube.com/watch?v=BlF60qYTrPI
Qualcomm FastCV SDK

- Similar API to OpenCV
- Collection of low-level computer vision primitives
- On many functions faster than OpenCV

http://www.youtube.com/watch?v=zTxpFCpG-GA
Preview: OpenVX – the new standard SDK

- OpenGL equivalent for hardware-accelerated computer vision
- Co-developed with OpenCV
- All major industry players
- First draft end 2013
Android SDK + Android NDK + OpenCV
Android SDK Installation

- Unzip the directory to a local directory *without* spaces in the path
Android NDK Installation

- Unzip to same top-level folder as the SDK
OpenCV for Android Installation

• Download from: http://opencv.org/downloads.html
• Unzip to same top-level folder as Android SDK and NDK
Set up Eclipse / ADT
Final directory layout

- After all SDKs are installed, the directory structure should look like similar to the screenshot below.

- Open the Eclipse folder and start the executable.
Start Eclipse / ADT
Start Android SDK Manager
Install the Android Support Library
Install the Android Support Library
Android Project Setup
Android Project Setup

- Right-click into the Eclipse project view
- Select Import.. – Existing projects into workspace
- Select OpenCV Folder
- All tutorials will show up in the project list
- Right click the Tutorial #2
Import new project
### Project Properties Setup – ENV variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>CWD</td>
<td>/Users/lomeier/src/android/OpenCV-2.4.2-android-sdk/samples/tutorial-3-native</td>
<td>BUILD SYSTEM</td>
</tr>
<tr>
<td>NDKROOT</td>
<td>/Users/lomeier/android/android-ndk-r18/</td>
<td>USER: CONFIG</td>
</tr>
<tr>
<td>PWD</td>
<td>/Users/lomeier/src/android/OpenCV-2.4.2-android-sdk/samples/tutorial-3-native</td>
<td>BUILD SYSTEM</td>
</tr>
</tbody>
</table>

- **CWD**: Current Working Directory
- **NDKROOT**: NDK Root Directory
- **PWD**: Project Working Directory

**Configuration**: Default (Active)
Commandline Setup

• Imported Eclipse projects can contain errors and fail to compile dependencies

• Set up your command line environment by adding this to your ~/.bashrc or ~/.bash_profile file:

```
export PATH=/usr/local/bin:/usr/local/sbin:/Users/lomeier/android/android-ndk-r8b:$PATH
export NDKROOT=/Users/lomeier/android/android-ndk-r8b
```

```
source ~/.bash_profile
```

• Make the ndk-build script executable

```
chmod u+x /Users/lomeier/android/android-ndk-r9/ndk-build.cmd
```
Commandline compilation

• Change to the folder to the project
  
  ```bash
  cd /Users/lomeier/src/android/OpenCV-2.4.6-
  android-sdk/samples/tutorial-3-native
  ```

• Force a ndk clean build
  
  ```bash
  ndk-build -B
  ```

• Build / upload now in Eclipse
Launch Application
Put Phone into developer mode

Settings -> About phone
Tap on build number

Settings -> Developer options
Enable USB debugging

Reset USB

UNPLUG USB.

RE-PLUG USB. CONFIRM THE USB CONNECTION.
Select Cell Phone
Troubleshooting

Unsatisfied link error?
Force a native build on the command line.

```
cd /dir/to/OpenCV/samples
cd tutorial-2-mixedprocessing
ndk-build -B
```
Run the Sample (jni/jni_part.cpp)

• Standard example code extracts FAST corners

Mat& mGr = *(Mat*)addrGray;
Mat& mRgba = *(Mat*)addrRgba;
vector<KeyPoint> v;

FastFeatureDetector detector(50);
detector.detect(mGr, v);
for( unsigned int i = 0; i < v.size(); i++ )
{
    const KeyPoint& kp = v[i];
    circle(mRgba, Point(kp.pt.x, kp.pt.y), 10, Scalar(255,0,0,255));
}
Institute of Visual Computing

Class Assignment (jni/jni_part.cpp)

- Extend sample to extract and match ORB features between two frames
- Then color all pixels with ORB features translucent blue

Mat& mGr = *(Mat*)addrGray; // This is the gray scale frame
Mat& mRgb = *(Mat*)addrRgba; // This is the frame in color
vector<KeyPoint> v; // You need to store the ones from the last frame
    // and this frame

OrbFeatureDetector detector(50);
detector.detect(mGr, v);
for( unsigned int i = 0; i < v.size(); i++ )
{
    const KeyPoint& kp = v[i];
    circle(mRgb, Point(kp.pt.x, kp.pt.y), 10, Scalar(0,0,255,100));
    line(... from old point to new match)
}

OrbDescriptorExtractor ...
BruteForceMatcher<cv::HammingLUT> ...
Class Assignment

• Hand in until next Thursday (October 3, 2013)
• Screenshot (from device, not photos of display)
• Code from jni_part.cpp, inlined into email

• [http://www.howtogeek.com/121133/how-to-take-screenshots-on-android-devices-since-4.0/](http://www.howtogeek.com/121133/how-to-take-screenshots-on-android-devices-since-4.0/)
Camera Calibration
Camera calibration

- Additional tutorial if required
- OpenCV functions available
- Undistortion and camera matrix map from image pixel locations to normalized camera coordinates ("rays")
Camera calibration

1. Find and match corners

2. Calibrate camera (undistortion and rectification)

3. Optimize matrix

4. Test result on image

bool findChessboardCorners(image, patternSize, corners, flags);

double calibrateCamera(objectPoints, imgPoints, imgSize, camMatrix, distCoeffs, rvecs, tvecs, flags, criteria);

Mat initCameraMatrix2D(objPoints, imgPoints, imgSize, aspectRatio);

void getOptimalNewCameraMatrix(camMatrix, distCoeffs, imgSize, alpha, newImgSize, PixROI, cPP);

void undistort(src, dst, cameraMatrix, distCoeffs, newCameraMatrix);
Project Plan and Management
Project Proposal Q&A

- Papers / Scope
- Library choice
- Milestones
- Block Diagram
- Gantt Chart

Computer Vision Lab

<MYTITLE>

Starting Point / Available Software
- Android SDK
- <other SDK>

Expected Results
- Main Functionality 1
- Main Functionality 2

Fallback Options
- Functionality / Aspect to leave out 1 if time runs out
- Functionality /Aspect to leave out 2

Documentation
- Wiki page covering the compilation and operation (has to be reproducible by the TA)
- 4 pages in IEEE paper format (http://www.ieee.org/conferences_events/conferences/publishingtemplates.html)

Milestones
Describe each milestone with 3-4 sentences

- M1: Initial toolset setup
- M2: First running software doing camera setup
- M3: First functionality working
- M4: Simple demo
- M5: Improvements
- M6: Wrap-Up
- M7: Final demo
Diagrams in Report

• Data flow / block diagram

• Gantt Chart

Week 1 | Week 2 | Week 3 | Week 4

| Work Package #1 | Task 1 |
| Work Package #2 | | Task 2 |
| Work Package #3 | Task 3 | Task 4 |
Project Management

- Use Github issue tracker and milestones [http://github.com](http://github.com)
- Use the Github for Windows / Mac OS app as GUI