

# OpenCV Performance on MAEMO

Iliya Mironov, Dmitry Mokeev  
Nizhniy Novgorod State University  
Nizhny Novgorod, Russia

## Abstract

OpenCV is an open source effective image processing library. This library expands image and video processing capabilities of Maemo platform and can be used by various applications that process embedded cameras images and videos as well as any other images/videos. The target of this project is analyzing the possibility of using OpenCV library with MAEMO devices and creating user application that use OpenCV. The tasks of our project are porting OpenCV to MAEMO, developing an example application that uses OpenCV, measuring resource consumption.

INDEX TERMS: OPENCV, LIBRARY, MOTION TRACKING, FACE RECOGNITION, MOTION TRACKING, MAEMO.

## I. INTRODUCTION

OpenCV (Open Source Computer Vision) [1, 3] is a library of programming functions for real time computer vision. OpenCV is released under a BSD license; it is free for both academic and commercial use. The library has >500 optimized algorithms (see figure below). It is used around the world, has >2M downloads and >40K people in the user group. Uses range from interactive art, to mine inspection, stitching maps on the web on through advanced robotics.

Augmentation of functionality and of mobile devices allows them to solve tasks before such as image/video processing and pattern recognition. High photo/video cameras resolution and increasing computing power mobile devices graphic acceleration provide the possibility to out some objects (e.g. faces,) in the picture, to do automatic photo processing.

The one of most effective and famous open source image processing libraries is OpenCV. It can perform many of transformations and calculations over images. This library is based on OpenGL and it allows you to improve efficiently of processing by using hardware capabilities of Maemo device.

## II. MAIN PART

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. In mid 2008, OpenCV obtained corporate support from Willow Garage, and is now again under active development. A version 1.1 "pre-release" was released in October 2008, and a book by two authors of OpenCV published by O'Reilly Media went on the market that same month.

This is indeed a powerful library that's totally used in the computer "vision", as well as the library finds its application in the work with graphics.

The simplicity and elegance of the library allows you to have a few minutes after they met to display a video image from the webcam.

The library is mainly written in C, which makes it portable to some specific platforms such as Digital signal processor. But wrappers for languages such as C# and Python have been developed to encourage adoption by a wider audience.

Basic modules library OpenCV: Cxcore - core libraries, CV - The module of image processing and computer vision, Highgui - Module I / O image and video, create the user interface [2]. These modules we used to work on the project.

<p style="text-align: center;"><b>Module Cxcore</b></p> <ul style="list-style-type: none"> <li>▶ Operations on Arrays</li> <li>▶ Dynamic Structures</li> <li>▶ Drawing Functions</li> <li>▶ XML/YAML Persistence</li> <li>▶ Clustering and Search in Multi-Dimensional Spaces</li> <li>▶ Utility and System Functions and Macros</li> </ul>	<p style="text-align: center;"><b>Module CV</b></p> <ul style="list-style-type: none"> <li>▶ Image Filtering</li> <li>▶ Geometric Image Transformations</li> <li>▶ Miscellaneous Image Transformations</li> <li>▶ Feature Detection</li> <li>▶ Motion Analysis and Object Tracking</li> <li>▶ Object Detection</li> </ul>
<p style="text-align: center;"><b>Module Highgui</b></p> <ul style="list-style-type: none"> <li>▶ User Interface</li> <li>▶ Reading and Writing Images and Video</li> </ul>	

We set the task of porting libraries OpenCV, we have implemented it for maemo4 maemo5. Tak as OpenCV is not covered by unit tests, and we had to check it is working manually. We tested the efficiency of the library on the following devices: n810 n900.

As a result of the work will create working debian packages, which will include a library and the tests that developers can use in their tests.

Since OpenCV was initially implemented and optimized for the characteristics of the processor x86, and we compile the library under the arm processor, we needed to verify the performance library uses a set of specific test cases, which are presented below . After all we will see, need to optimize or build upon the library given the particular arm processor.

We turn to case tests that we chased the devices.

### III. TEST CASES

We have selected the following tests to verify the performance OpenCV libraries on devices MAEMO.

Tests were conducted on these devices: PC (Intel Pentium M: 1.7MHz, RAM: 1GB), N810 (Texas Instruments OMAP 2420, 400 MHz, DDR RAM 128 Mb), N900 (ARM Cortex-A8 CPU 600 MHz, RAM: 256 Mb, Swap: 768 Mb).

#### A. Test Cases Linear Filtering (Fig. 1)

The first test we checked the speed of the code that implements a linear filtering. So we can estimate the speed of the processor when the library OpenCV.

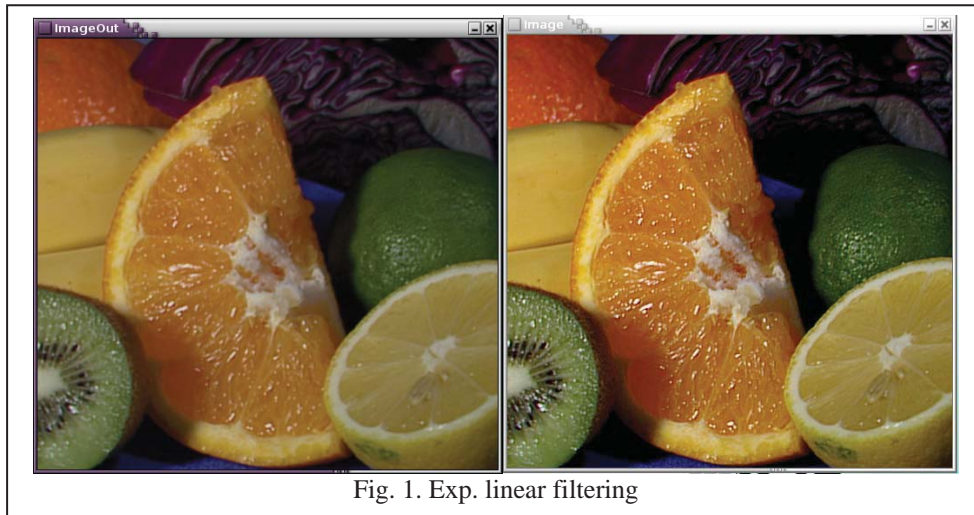


Fig. 1. Exp. linear filtering

### B. Test Cases Face recognition (Figure 2)

The second test checks the speed of the code which carried out to find the person in the picture. When using this test used built-in features and classifiers library OpenCV



Fig. 2. Exp. Face recognition

### C. Test Cases Motion tracking (Figure 3)

In the third test is for the possibility of using a video capture library OpenCV, as well as assess the possibility of simultaneous processing of video processing and transmission of video in real time.

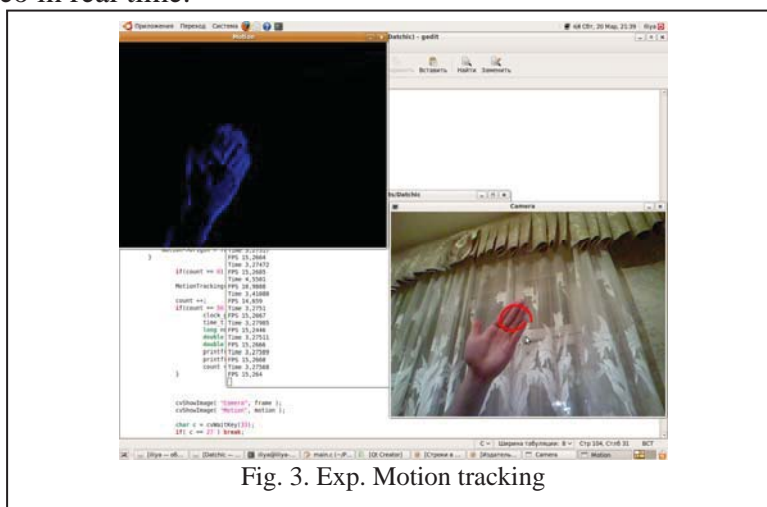
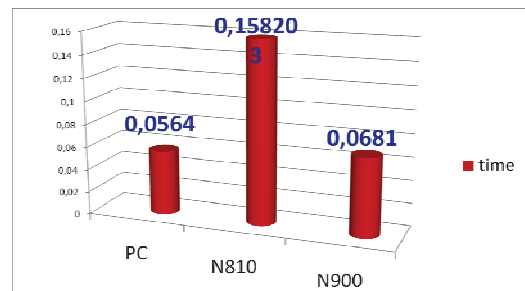


Fig. 3. Exp. Motion tracking

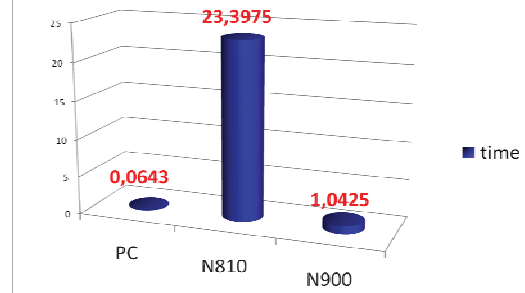
## IV. RESULTS OF PERFORMANCE TESTS

**Test case 1 – linear filtering**

time measured in seconds

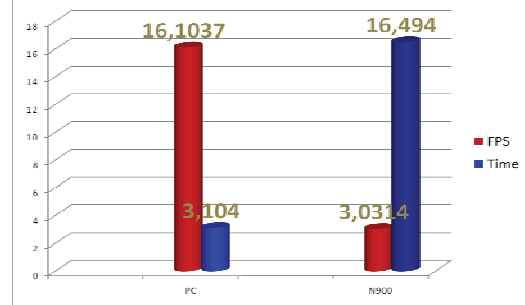
**Test case 2 – face recognition**

time measured in seconds

**Test case 3 – motion tracking**

time measured in seconds

fps - frames per second



## V. CONCLUSIONS

OpenCV is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Mac OS X, Windows and Linux. It focuses mainly towards real-time image processing, as such, if it finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself.

We perform the build OpenCV library for maemo4 maemo5, running on devices Maemo N810, N900. Because the patched OpenCV was implemented and optimized for the characteristics of the processor x86, and we all compile on a workstation, we tested its performance using several different test cases which are explained above.

As a result of testing shows that require further study optimize or build upon the library given the particular processor arm.

## REFERENCES

- [1] <http://opencv.willowgarage.com/wiki/>
- [2] <http://opencv.willowgarage.com/documentation/cpp/introduction.html>
- [3] Bradsky G., Kaehler A. Learning OpenCV - O'Reilly, 2008. - C. 1