

*The most trusted source of analysis, advice, and engineering
for embedded processing technology and applications*



***Get up and Running Quickly With Embedded Vision
Using OpenCV on Android
Eric Gregori***

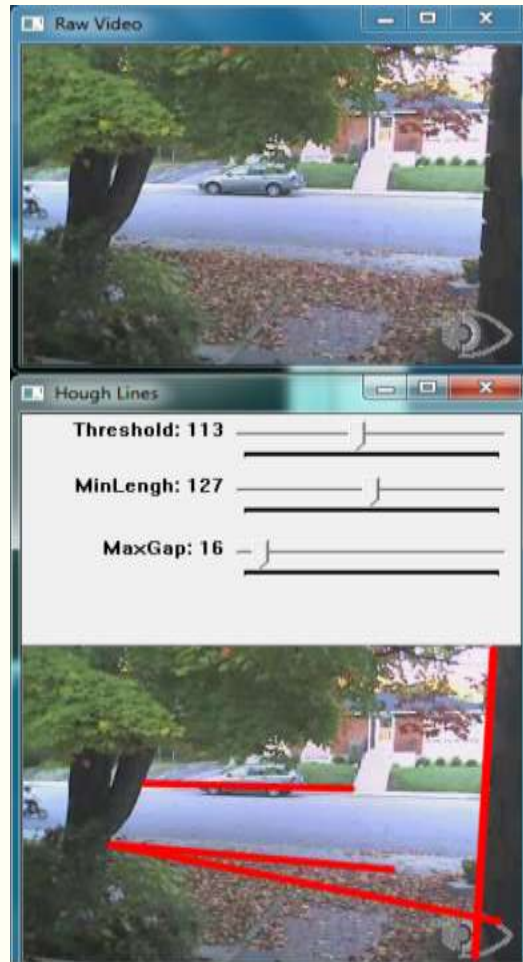
ARM TechCon 2012—Santa Clara

Berkeley Design Technology, Inc.
Oakland, California USA
+1 (510) 451-1800

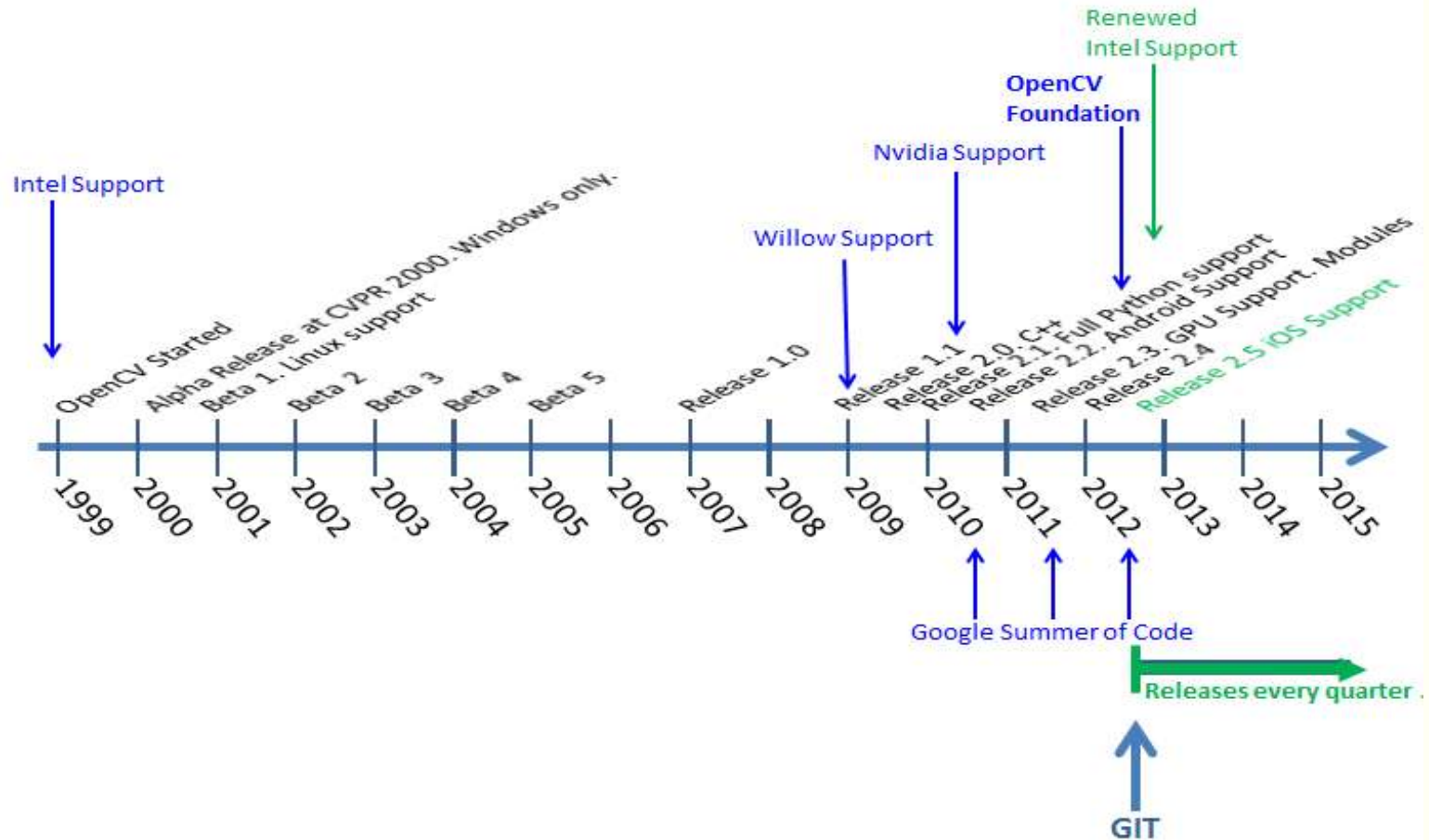
info@BDTI.com
<http://www.BDTI.com>



What is Embedded Vision?



OpenCV Timeline

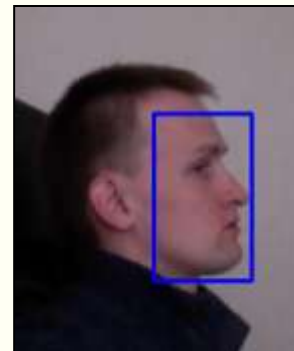


Courtesy of Gary Bradski

Recent Functionality in OpenCV

Major New Functionality

- Face recognition (contributed by Philipp Wagner)
- FREAK keypoint descriptor (from EPFL lab)
- GMG background subtractor (contributed by A. B. Godbehere)
- Video stabilization module (by OpenCV NVIDIA team)
- Enhanced LogPolar transform
- OpenFABMAP image recognition algorithm (for image retrieval)
- Better solvePnP algorithms 2D points to 3D pose (implementations of EPFL algorithms)
- From Google Summer of Code
 - Image denoising
 - Dense optical flow
 - 2 new object detectors
 - OpenCV iOS port
 - More C++ samples
 - Python samples



Cascade: Side face, and silverware

Courtesy of Gary Bradski

What Can OpenCV Do?



Image
Processing



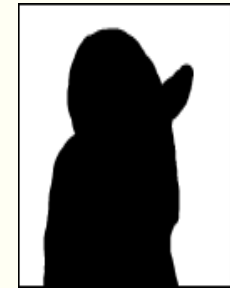
Transforms



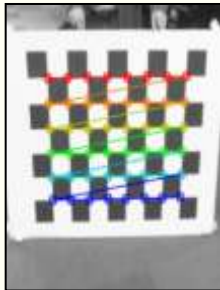
Fitting



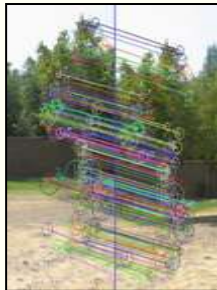
Optical Flow
Tracking



Segmentation



Calibration



Features
VSLAM



Depth, Pose
Normals,
Planes, 3D
Features



Object
Recognition
Machine
Learning



Computational
Photography

Courtesy of Gary Bradski

Where is OpenCV Used?

- Academic and industry research
- Security systems
- Google Maps, Streetview
- Image/video search and retrieval
- Structure from motion in movies
- Machine vision factory production inspection systems
- Automatic driver assistance systems
- Safety monitoring (dam sites, mines, swimming pools)
- Robotics



2M downloads

Courtesy of Gary Bradski

The New Face of OpenCV: OpenCV.org

OpenCV

ABOUT
 DOWNLOADS
 DOCUMENTATION
 PLATFORMS
 SUPPORT
 CONTRIBUTE

SEARCH

DONATE

OPENCV
 (OPEN SOURCE
 COMPUTER VISION)

OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Android and Mac OS. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multicore processing. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 5 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

QUICK LINKS:

- [Online documentation](#)
- [User Q&A forum](#)
- [Report a bug](#)
- [Developers zone](#)
- [Build farm](#)

LATEST DOWNLOADS

04/07/2012
VERSION 2.4.2

- [OpenCV for Windows](#)
- [OpenCV for Linux/Mac](#)
- [OpenCV for Android](#)
- [OpenCV for iOS](#)

WHAT'S NEW

Date	Title
06/10/2012	OpenCV week in Italy This week in Italy two OpenCV related events
25/07/2012	Migration to git. 80% completed... Greetings! We're glad to
04/07/2012	OpenCV v2.4.2 released It should be binary compatible with OpenCV
13/06/2012	OpenCV Hits Five Million! Today OpenCV crosses

OpenCV in The Embedded Space

OpenCV has always been available to the embedded space under Linux

The library has been ported to: PowerPC, MIPS, Blackfin, Xscale and ARM

If it can run Linux, it can run OpenCV



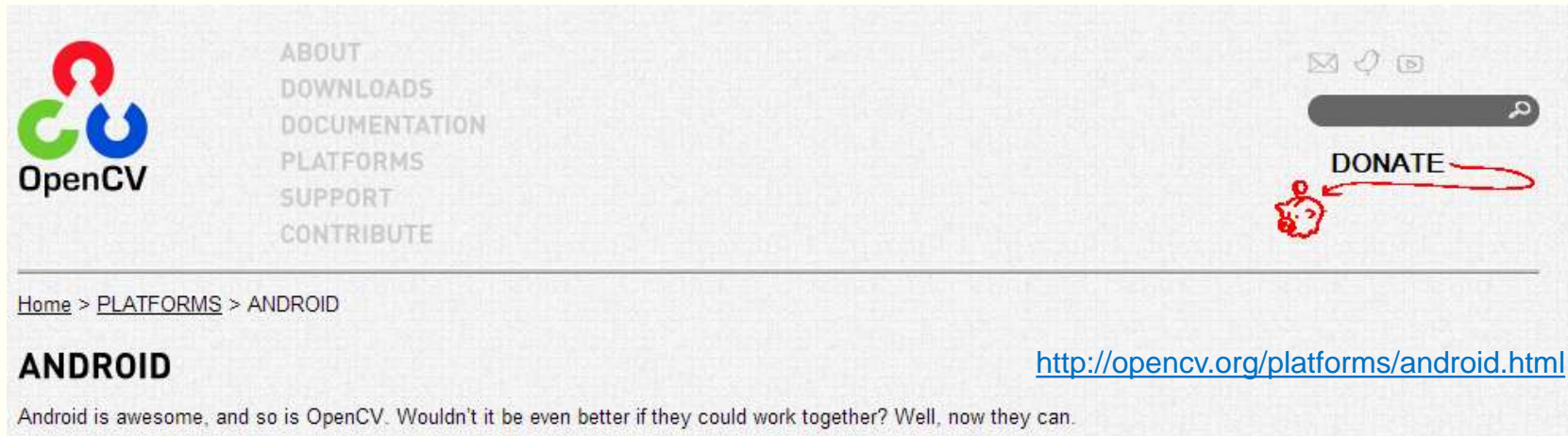
TI BeagleBoard
TI BeagleBone
Raspberry Pi (Broadcom)
Analog Devices Blackfin
Freescale i.MX
Android (Qualcomm, NVIDIA)
iOS



<http://whatnicklife.blogspot.com/2010/05/beagle-has-2-eyes-opencv-stereo-on.html>



OpenCV4Android



OpenCV 2.4 for Android:

- Native Android camera support
- Multithreading
- Java API
- Tegra hardware optimizations
- OpenCV Manager



Courtesy of Gary Bradski

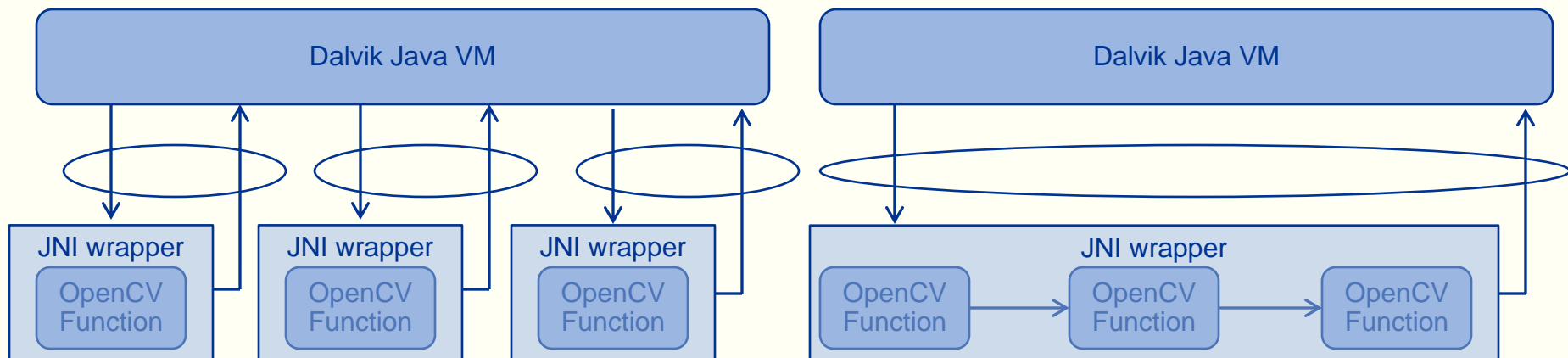
OpenCV4Android Development Java or C++ or Both

Java (basic)

- The Android way
- OpenCV Java API (wrappers)
- Computations are performed on a native level
- JNI call overhead
- Multiple JNI calls in pipeline

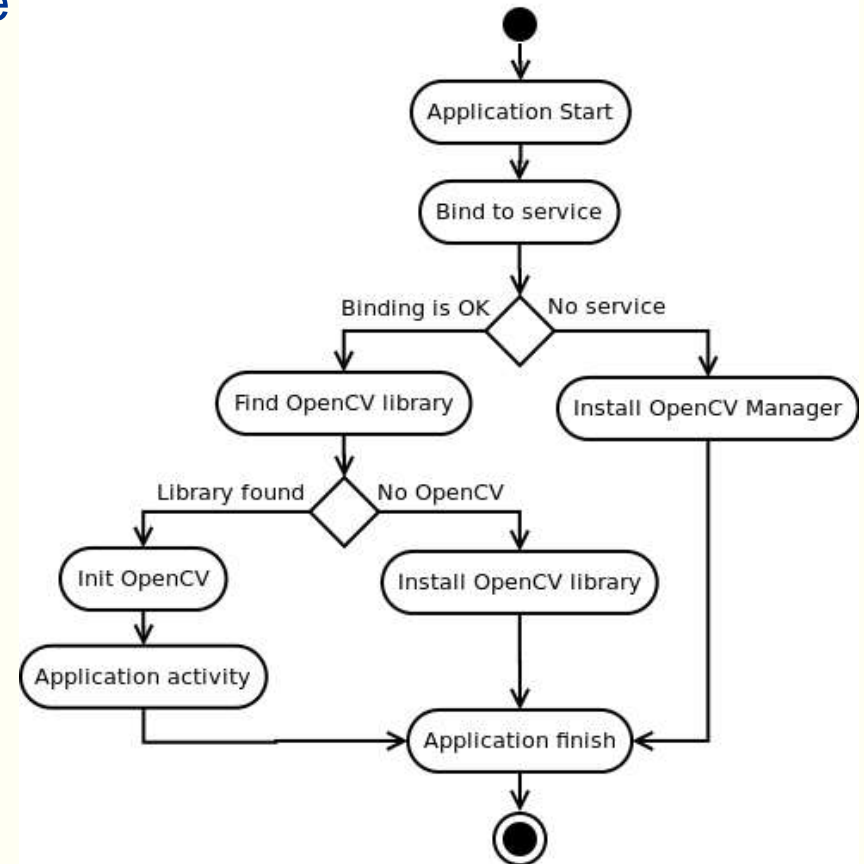
Native C++

- JNI—Java Native Interface
- Native C++ OpenCV API
- Fewer JNI calls, faster performance
- One JNI call for pipeline
- Easy port from Desktop



OpenCV Manager

- Android service targeted to manage OpenCV library binaries on end user devices
- Allows sharing the OpenCV dynamic libraries of different versions between applications on the same device
- Installed and updated from Google Play
- Guarantees usage of current/trusted OpenCV libraries
- Less memory usage



HOW TO BUILD EMBEDDED-VISION APPLICATIONS USING OPENCV ON ANDROID

Best-in-class On-line Documentation/Tutorials



The image shows three screenshots of OpenCV documentation and DevZone pages. The top two screenshots are tutorial pages for using OpenCV4Android SDK with Eclipse and using OpenCV in C++ code with OpenCV4Android SDK. The bottom screenshot is the OpenCV DevZone page for OpenCV, showing a sidebar with navigation options (Overview, Activity, Roadmap) and a main content area with a WikiStart link to OpenCV4Android and a section titled 'Building OpenCV4Android from trunk'.

OpenCV v2.4.9 documentation » OpenCV Tutorials » Introduction to OpenCV » [previous](#) | [next](#) | [index](#)

Using OpenCV4Android SDK with Eclipse

This tutorial was tested using Ubuntu 10.04 and Windows 7 SP1 operating systems. However, it should also work with any other OS, supported by Android SDK. If you encounter any error after thoroughly following these steps, feel free to contact us via [OpenCV4Android](#) discussion group or [OpenCV Q&A forum](#). We'll do our best to help you out.

OpenCV v2.4.9 documentation » OpenCV Tutorials » Introduction to OpenCV » [previous](#) | [next](#) | [index](#)

Using OpenCV in C++ code with OpenCV4Android SDK

The Android way is writing all your code in Java. But sometimes, it is not enough and you need to go to the native level and write some parts of your application in C/C++. This is especially important when you already have some computer vision code which is written in C++ and uses OpenCV, and you want to reuse it in your Android application. In this case the only way is to use JNI - a Java framework for interaction with native code. It means, that you should add a Java class with native methods exposing your C++ functionality to the Java part of your Android application.

OpenCV DevZone [Projects](#) [?](#) [Register](#) [Sign in](#)

OpenCV

- Overview
- Activity
- Roadmap

WikiStart » OpenCV4Android »

Building OpenCV4Android from trunk

The new android build is based on android-cmake project originally developed by Ethan Rublee. You can find the latest version of his project at <http://code.google.com/p/android-cmake/>

Contents

[Developing OpenCV Applications Using the Java API](#)

[Developing OpenCV Applications Using the Native API \(C++\)](#)

[Building the Android OpenCV Libraries From Source](#)

Tools Required for OpenCV Android Development

- Android
 - Sun/Oracle JDK 6
 - Android NDK
 - Android SDK and components
 - Android SDK Tools, revision14 or newer
 - SDK Platform Android 3.0, API 11 (also known as android-11)
 - Eclipse IDE
 - ADT plugin for Eclipse
- OpenCV
 - [OpenCV-2.4.2-android-sdk.zip](#)

Installing Android the Easy Way—TADP



Home

TEGRA ANDROID DEVELOPMENT PACK

Setting up an Android development environment can be a complex and frustrating experience. NVIDIA simplifies this for all Android developers with a single installer that manages this complexity for you. The Tegra Android Development Pack installs all software tools required to develop for Android on NVIDIA's Tegra platform and is the perfect companion for developing native applications for Tegra DevKits. This suite of developer tools is targeted at Tegra devices, but will configure a development environment that will work with almost any Android device.

- The Tegra Android Development Pack—TADP makes installing the Android development tools automatic
- TADP can be used even if you are building for an Android device that does NOT use a Tegra application processor
- TADP installers are available for:
 - Windows
 - OSX
 - Ubuntu 32bit —Requires Java
 - Ubuntu 64bit —Requires Java
- Development Tools Included:
 - NVIDIA Debug Manager for Eclipse 14.0.1
 - Android SDK r18
 - Android NDK r8
 - JDK 1.6.0_24
 - Cygwin 1.7
 - Eclipse 3.7.1
 - CDT 8.0.0
 - ADT 18.0.0
 - Apache Ant 1.8.2

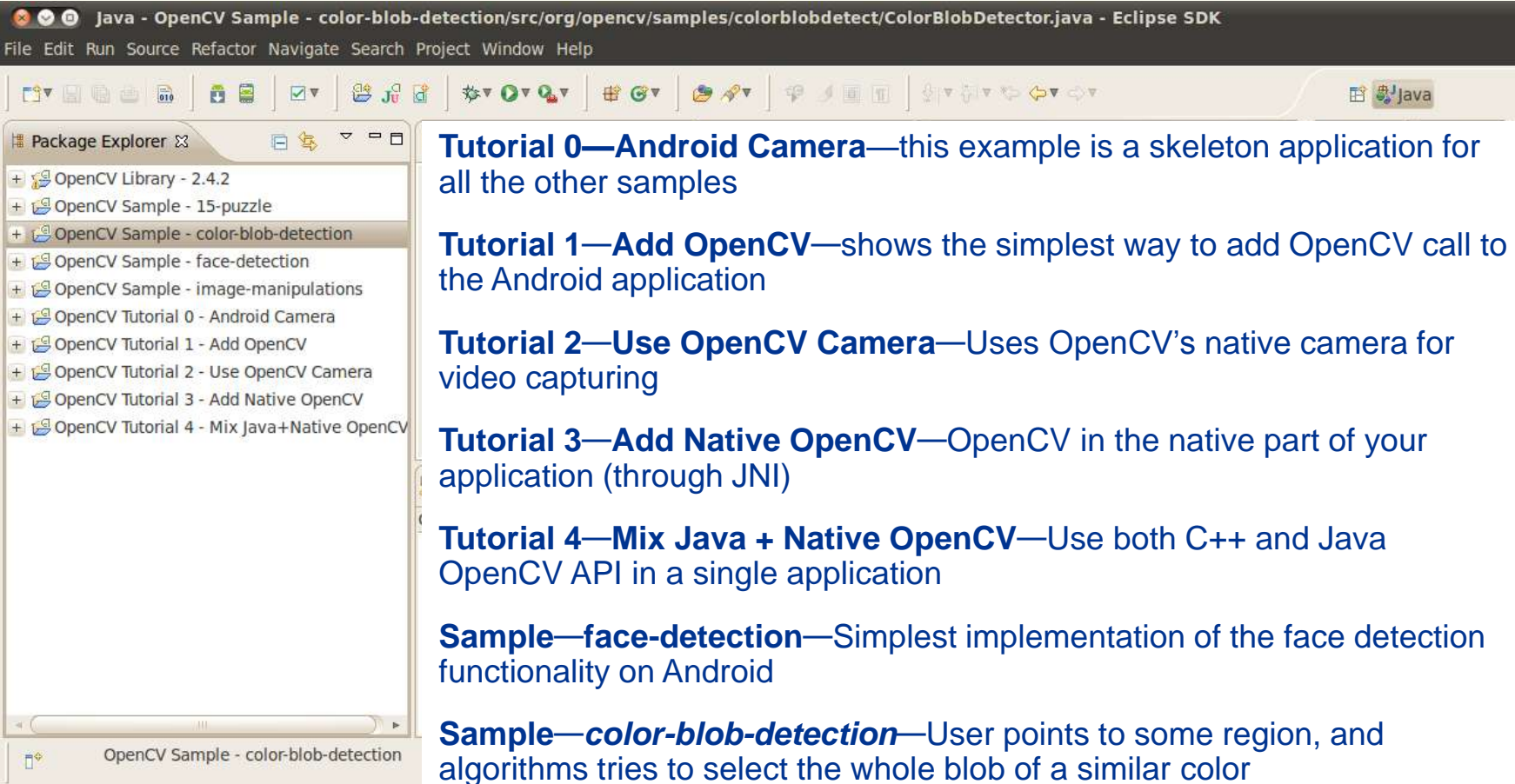
Creating a Build Environment on Ubuntu 10.04 64bit

Using OpenCV4Android SDK with Eclipse

This tutorial was tested using Ubuntu 10.04 and Windows 7 SP1 operating systems. Nevertheless, it should also work with any other **OS**, supported by Android SDK (including Mac OS X). If you encounter errors after following the steps described here, feel free to contact us via [OpenCV4Android](#) discussion group or OpenCV [Q&A forum](#) and we will try to help you.

- 1) 64bit Ubuntu is NOT required for application building, but is required if you decide to build the Android OS
- 2) Install Ubuntu JRE (used for TADP installation)
- 3) Download and install TADP for Ubuntu 64bit
- 4) * Install ia32 shared libraries: `sudo apt-get install ia32-libs`
- 5) * `gedit ~/.profile`
 - `export PATH=$PATH:/home/bdti/NVPACK/android-ndk-r8`
 - Log out then log in
- 6) Download the OpenCV Android SDK: `OpenCV-2.4.2-android-sdk.zip`, unzip into the Ubuntu file system
- 7) Import OpenCV projects into Eclipse installed by TADP
- 8) * Edit face detection sample and tutorial's 3,4 C++ properties
 - Build command: `${NDKROOT}/ndk-build`

Installed and Ready to Start Development



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left lists the following items:

- OpenCV Library - 2.4.2
- OpenCV Sample - 15-puzzle
- OpenCV Sample - color-blob-detection
- OpenCV Sample - face-detection
- OpenCV Sample - image-manipulations
- OpenCV Tutorial 0 - Android Camera
- OpenCV Tutorial 1 - Add OpenCV
- OpenCV Tutorial 2 - Use OpenCV Camera
- OpenCV Tutorial 3 - Add Native OpenCV
- OpenCV Tutorial 4 - Mix Java+Native OpenCV

The main editor area is currently empty. The title bar of the IDE window reads: "Java - OpenCV Sample - color-blob-detection/src/org/opencv/samples/colorblobdetect/ColorBlobDetector.java - Eclipse SDK".

Tutorial 0—Android Camera—this example is a skeleton application for all the other samples

Tutorial 1—Add OpenCV—shows the simplest way to add OpenCV call to the Android application

Tutorial 2—Use OpenCV Camera—Uses OpenCV’s native camera for video capturing

Tutorial 3—Add Native OpenCV—OpenCV in the native part of your application (through JNI)

Tutorial 4—Mix Java + Native OpenCV—Use both C++ and Java OpenCV API in a single application

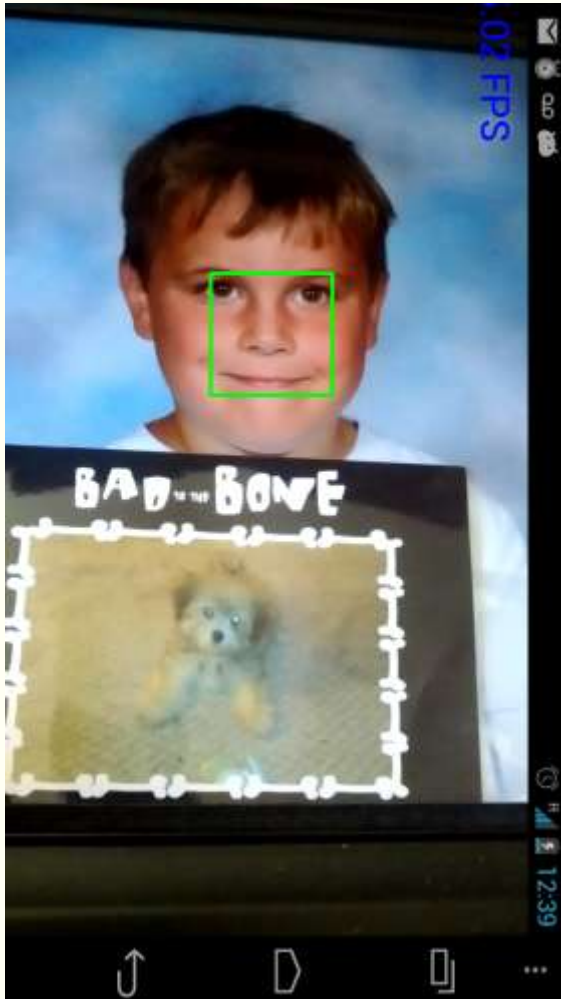
Sample—face-detection—Simplest implementation of the face detection functionality on Android

Sample—*color-blob-detection*—User points to some region, and algorithms tries to select the whole blob of a similar color

SHOW DEMOS HERE

FACE DETECTION

Face Detection



Face detection is one use of an algorithm that is trained to look for specific features, in a specific order.

Instead of being programmed, this algorithm learns what an object looks like through training.

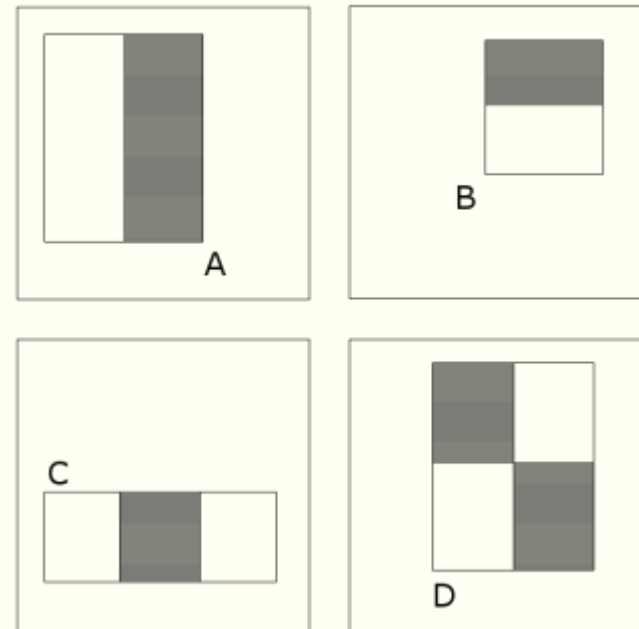
Training is done offline, and is accomplished by “showing” the learning algorithm both positive and negative images (images with a face and without a face).

Face Detection—Using Haar Features

Four distinct templates referred to as **Haar features**.

Templates can be processed faster than other techniques.

The template is laid over a portion of the image, and a weight is calculated based on the pixels under the template.

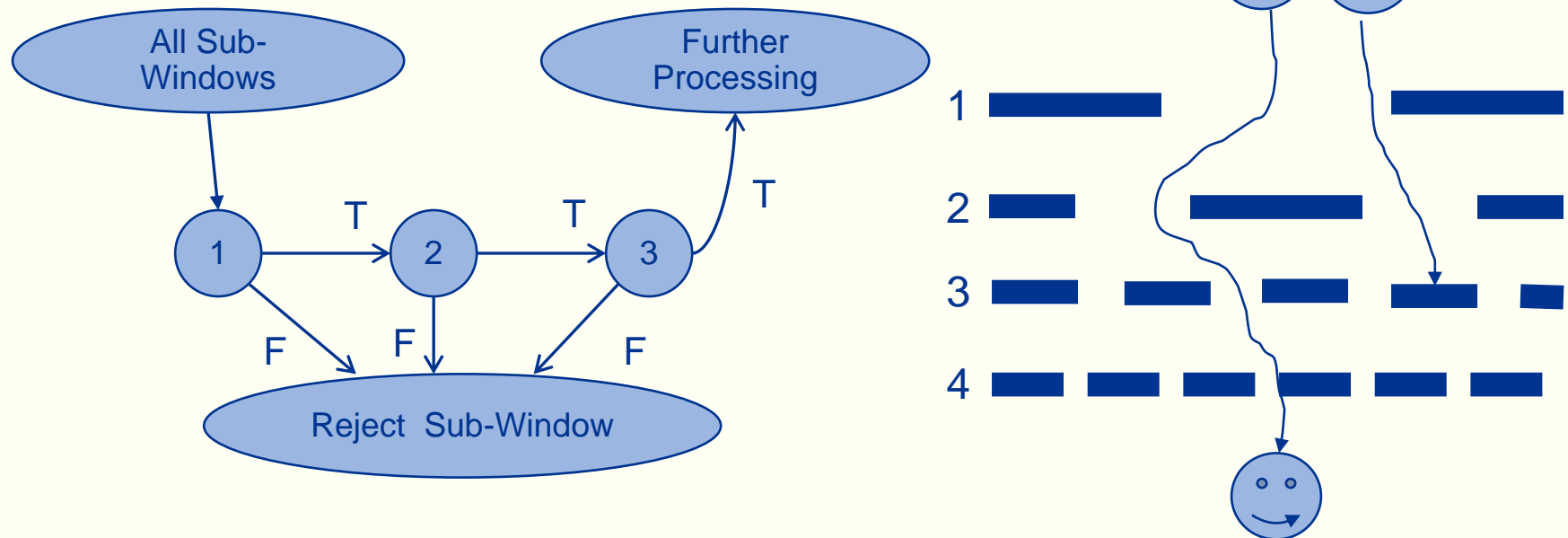


Face Detection

How does training work?

A face of 24 24 pixels can have 45,396 possible combinations/scales of the templates from the pervious slide.

The purpose of training is to reduce the 45,396 possible combinations down to a minimum number and an ideal order.

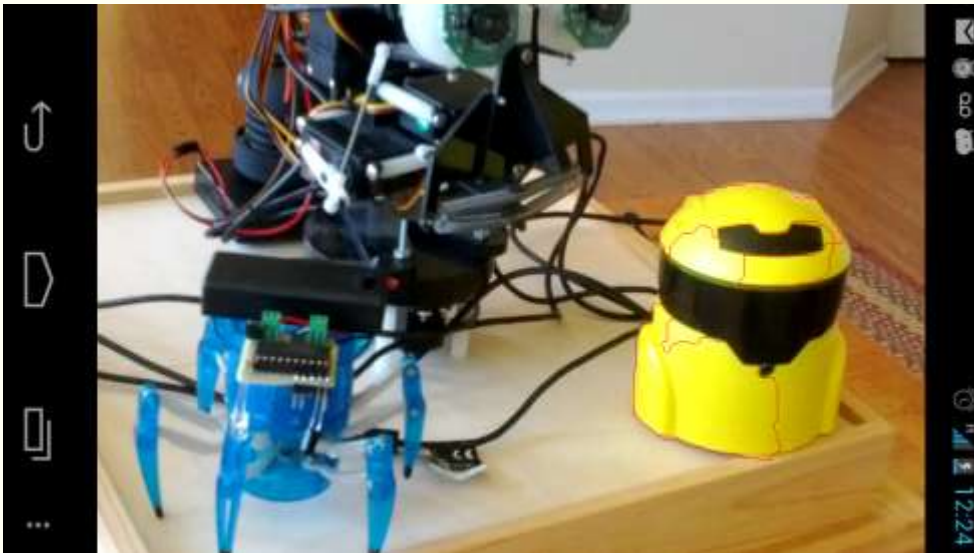


COLOR BLOB DETECTION / CONTOURS

Color Blob Tracking

A “color blob” is a group of adjacent pixels with a common color component.

Segmenting objects based on color is a very efficient method of separating foreground objects from background objects.



Works well if object of interest is a distinct color.

Problem: Camera “sees” color changes with lighting due to limited dynamic range.

Color Blob Tracking



Contours are chains of similar connected features defining a line/curve in an image.

A contour associates many individual features into a single segment.

Many individual features (yellow pixels).

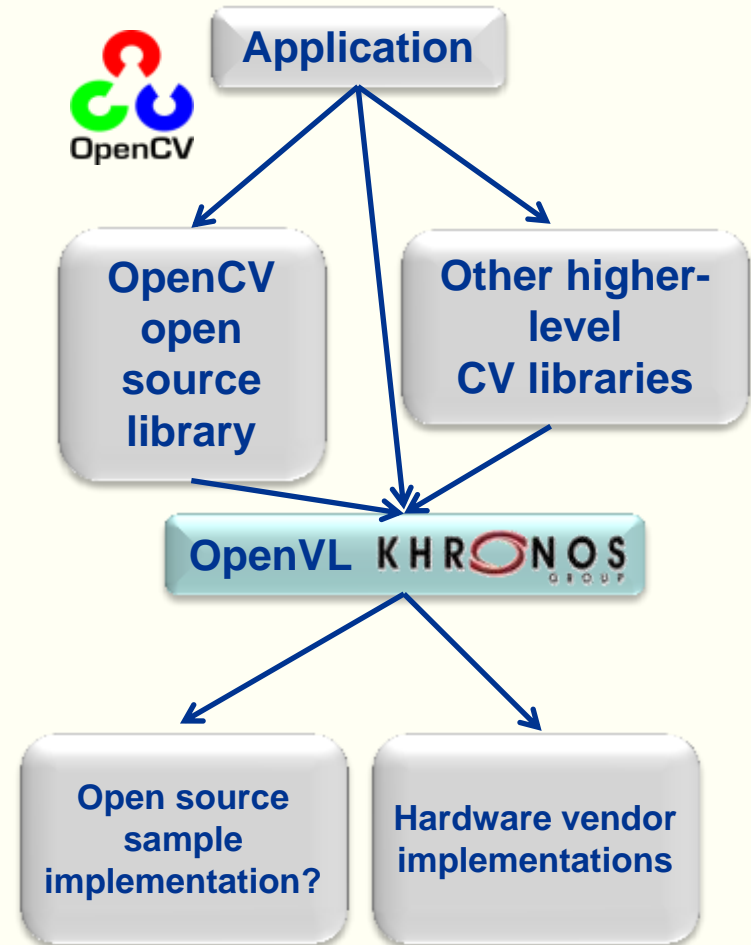
Single segment (defined by red line).

Contour only defines boundary, not content (not all pixels in segment are yellow).

THE FUTURE OF OPENCV

OpenCV Helping Drive the New Khronos Standard: OpenVL

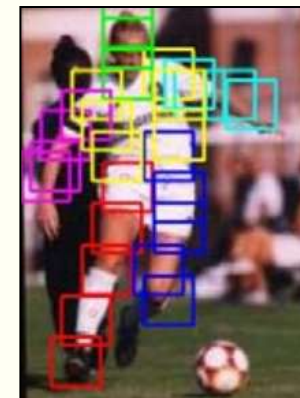
- Vision Hardware Acceleration Layer
 - Enable hardware vendors to implement accelerated imaging and vision algorithms
- OpenVL can be used by high-level libraries or applications directly
 - Primary focus on enabling real-time vision apps on mobile and embedded systems
- Future versions of OpenCV will leverage OpenVL
- Working group aiming for stable draft spec in 2012



Courtesy of Gary Bradski

Coming Highlights in OpenCV

- Faster releases 4x-6x/year
- Cloud support (python on Amazon servers)
- Revamped mathematical framework for detectors and descriptors:
 - Faster and way more accurate
- Depth motion fusion
- Iris Recognition
- Transparent item ID
- ARM optimization(?)
- 3D model training
- 2D barcodes
- 2D line matching
- Parts from whole
- More modular
- More optimized



User: <http://opencv.org>

Developer: <http://code.opencv.org>

Courtesy of Gary Bradski

Summary

- Embedded vision enables systems to “see and understand” their environments, making them more intelligent and responsive
- OpenCV is a popular, free computer vision library supported by industry and academia. It supports over 2500 algorithms and has been downloaded over 5 million times
- With the help of NVIDIA, OpenCV has been ported to the Android operating system
- Using the Google Android tools and NVIDIA installer, developing OpenCV applications on Android is easy

RESOURCES

Selected Resources: The Embedded Vision Alliance

The Embedded Vision Alliance is an industry partnership to transform the electronics industry by inspiring and empowering engineers to design systems that see and understand



Free Resources from the Embedded Vision Alliance

The Embedded Vision Alliance web site, at www.Embedded-Vision.com covers embedded vision applications and technology, including interviews and demonstrations



The Embedded Vision Academy, a free service of the Alliance, offers free in-depth tutorial articles, video “chalk talks,” code examples and discussion forums: www.EmbeddedVisionAcademy.com



The Embedded Vision Insights newsletter provides updates on new materials available on the Alliance website. Sign up at www.Embedded-Vision.com/user/register



Embedded vision technology and services companies interested in becoming sponsoring members of the Alliance may contact info@Embedded-Vision.com



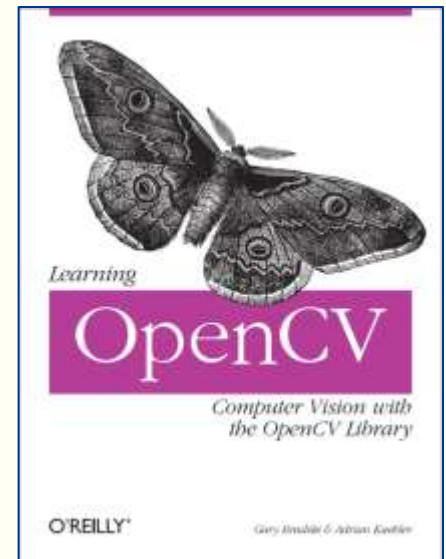
Useful OpenCV Links

- **Developer OpenCV Site:** <http://code.opencv.org>
- **User OpenCV Site:** <http://opencv.org>
- **User Group:**
<http://tech.groups.yahoo.com/group/OpenCV/join>
- **Book on OpenCV:**
 - <http://oreilly.com/catalog/9780596516130/>
 - <http://www.amazon.com/Learning-OpenCV-Computer-Vision-Library/dp/0596516134>
 - **Code examples from the book:**
<http://examples.oreilly.com/9780596516130/>

Version 2 of the book is coming Q4, 2012

For high level issues, partnering, financial contributions, consulting, contract services:

Contact: garybradski@gmail.com



Best seller in Computer Vision and Machine Learning for 3 years. Version 2 coming this summer.

Additional Resources

BDTI's web site, www.BDTI.com, provides a variety of free information on processors used in vision applications.



BDTI's free "InsideDSP" email newsletter covers tools, chips, and other technologies for embedded vision and other DSP applications. Sign up at www.BDTI.com.

