OpenCV 3.0
Uses in Robotics and AR

Gary Bradski

VP Perception and Core Software,
Magic Leap

Director: OpenCV Foundation

OpenCV Thanks!
For Key Support

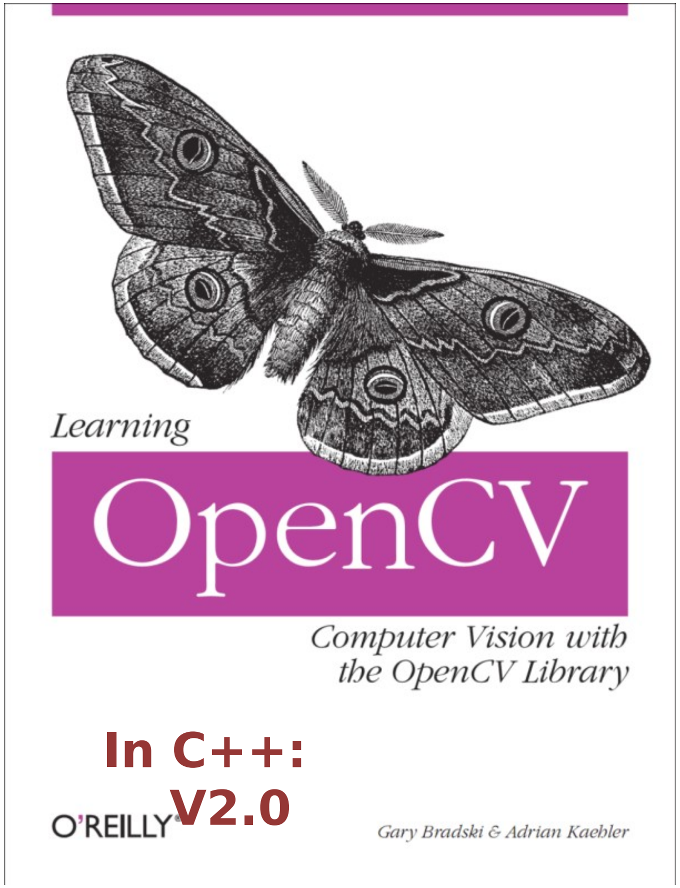
- Intel
  - for getting it started and helping fund the challenge
- Google
  - for growing support in Google Summer of Code over the last 5 years
- Nvidia
  - Supporting Cuda version with lots of other help

Outline: OPENCV 3.0

- **Intro**
  - Learning OpenCV Version 2.0 coming by Aug
  - Announcing $50K Vision Challenge
- **OpenCV Background**
- **OpenCV 3.0 High Level**
- **OpenCV 3.0 Modules**
- **Brand New in OpenCV**
- **OpenCV Examples**
  - Robotics
  - Augmented Reality

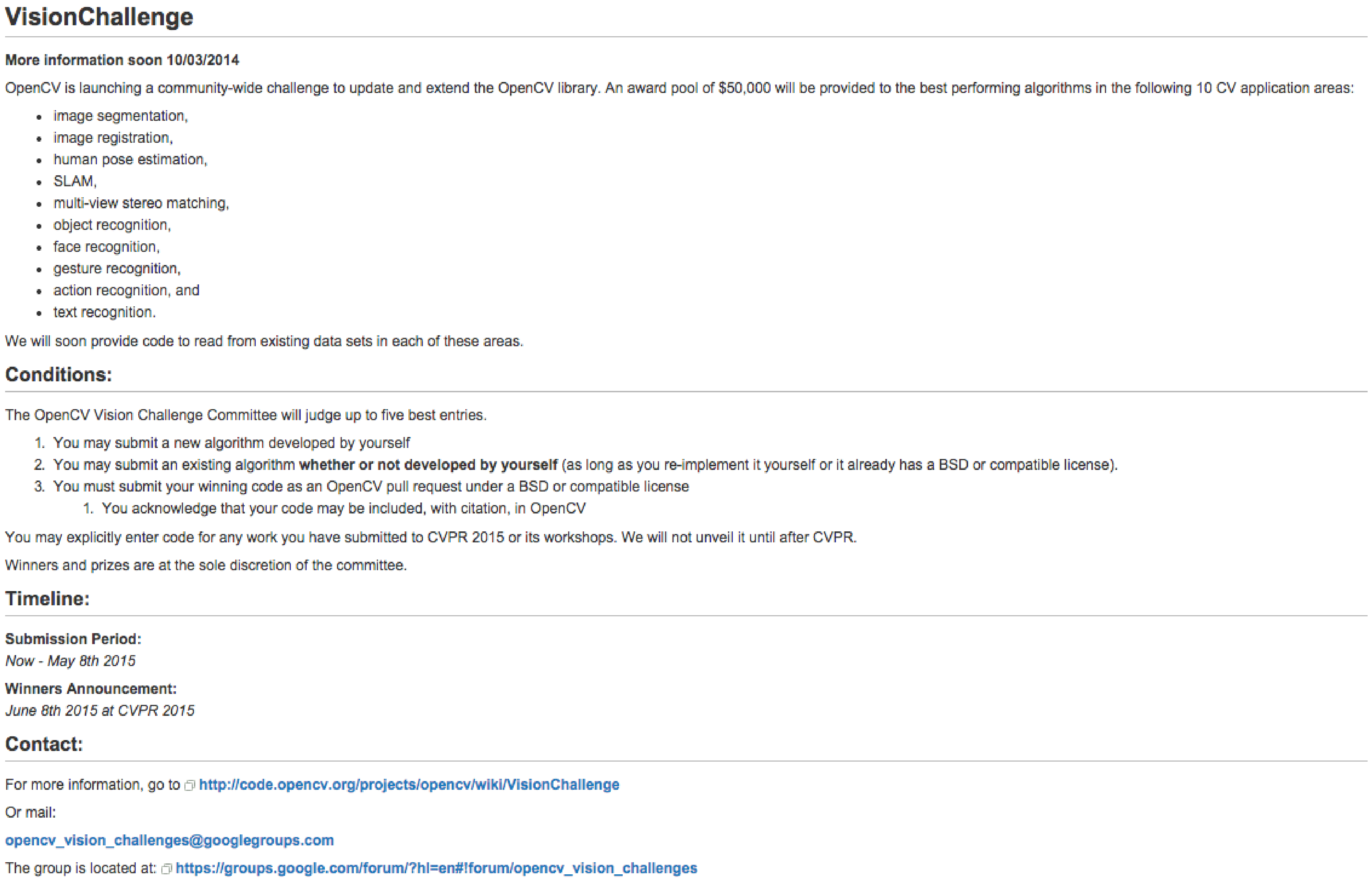
Learning OpenCV V2.0

- Out in Summer 2014!


Learning
OpenCV
Computer Vision with the OpenCV Library
O'REILLY
Gary Bradski & Adrian Kaehler

In C++: V2.0

OpenCV $50K Vision Challenge



VisionChallenge

More information soon 10/03/2014

OpenCV is launching a community-wide challenge to update and extend the OpenCV library. An award pool of $50,000 will be provided to the best performing algorithms in the following 10 CV application areas:

- image segmentation,
- image registration,
- human pose estimation,
- SLAM,
- multi-view stereo matching,
- object recognition,
- face recognition,
- gesture recognition,
- action recognition, and
- text recognition.

We will soon provide code to read from existing data sets in each of these areas.

Conditions:

The OpenCV Vision Challenge Committee will judge up to five best entries.

1. You may submit a new algorithm developed by yourself
2. You may submit an existing algorithm **whether or not developed by yourself** (as long as you re-implement it yourself or it already has a BSD or compatible license).
3. You must submit your winning code as an OpenCV pull request under a BSD or compatible license
    1. You acknowledge that your code may be included, with citation, in OpenCV

You may explicitly enter code for any work you have submitted to CVPR 2015 or its workshops. We will not unveil it until after CVPR.

Winners and prizes are at the sole discretion of the committee.

Timeline:

Submission Period:
Now - May 8th 2015

Winners Announcement:
June 8th 2015 at CVPR 2015

Contact:

For more information, go to http://code.opencv.org/projects/opencv/wiki/VisionChallenge

Or mail:

opencv_vision_challenges@googlegroups.com

The group is located at: https://groups.google.com/forum/?hl=en#!forum/opencv_vision_challenges

OpenCV Background

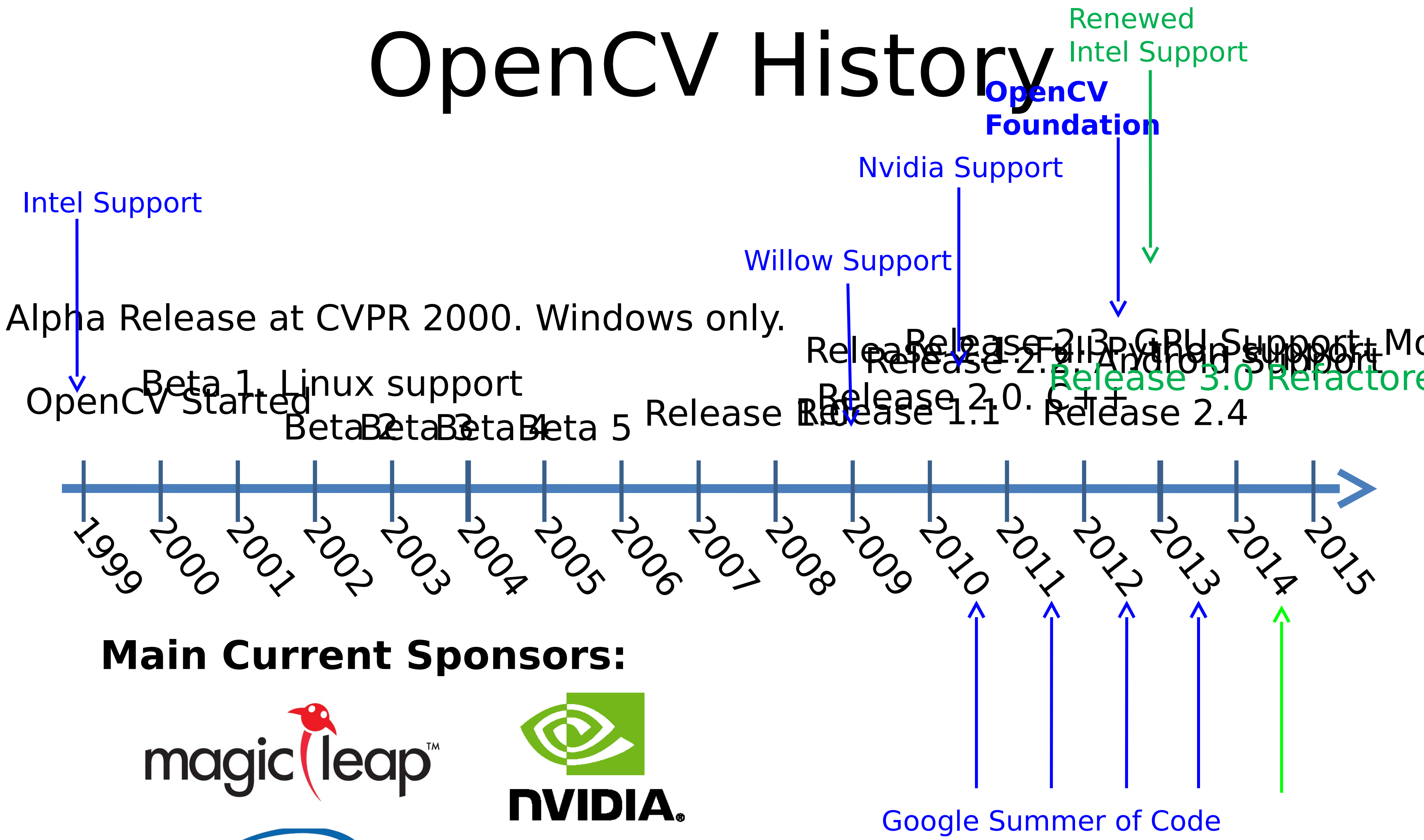
What is OpenCV

- **Open** Source **C**omputer **V**ision Library
- Routines focused on real time image processing and 2D + 3D computer vision.
  - **On Linux, Windows, Mac, Android and iOS**
  - **C++, C, Java, Matlab and Python interfaces**
- **Free** for commercial or research use in whole or in part.

OpenCV License

- Based on BSD license
- Free for **commercial** and research use
- Does **not force** your code to be open
- You need not contribute back
  - But you are very welcome to contribute back!

OpenCV History



Main Current Sponsors:

magic leap™

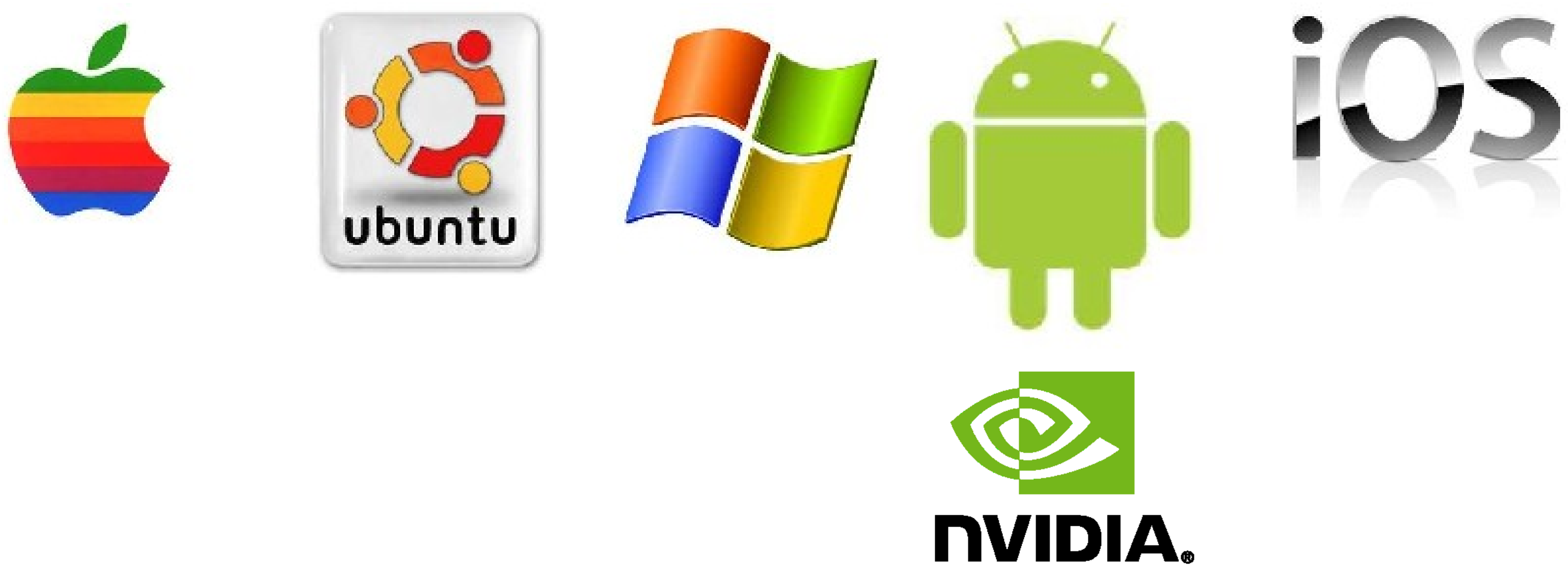
NVIDIA®

intel®

Google™

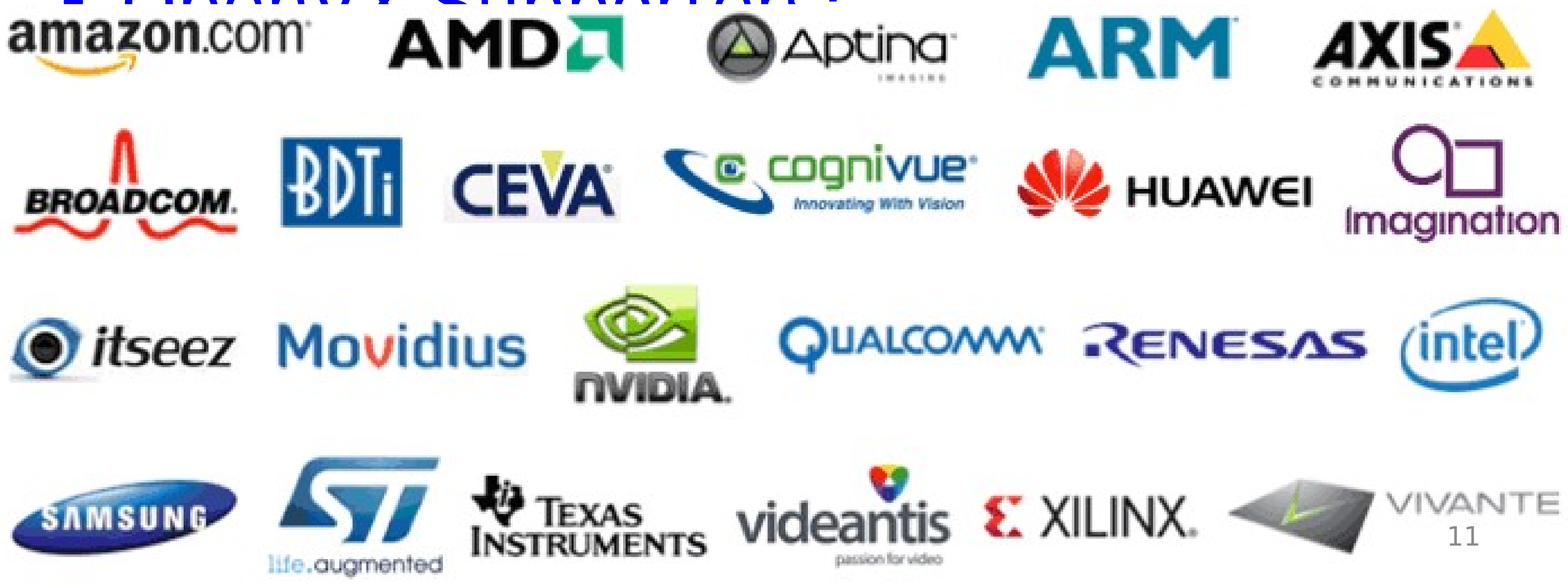
Environments, Platforms

- Languages:
  - C++, C#, Python, C, Java
- Platforms:

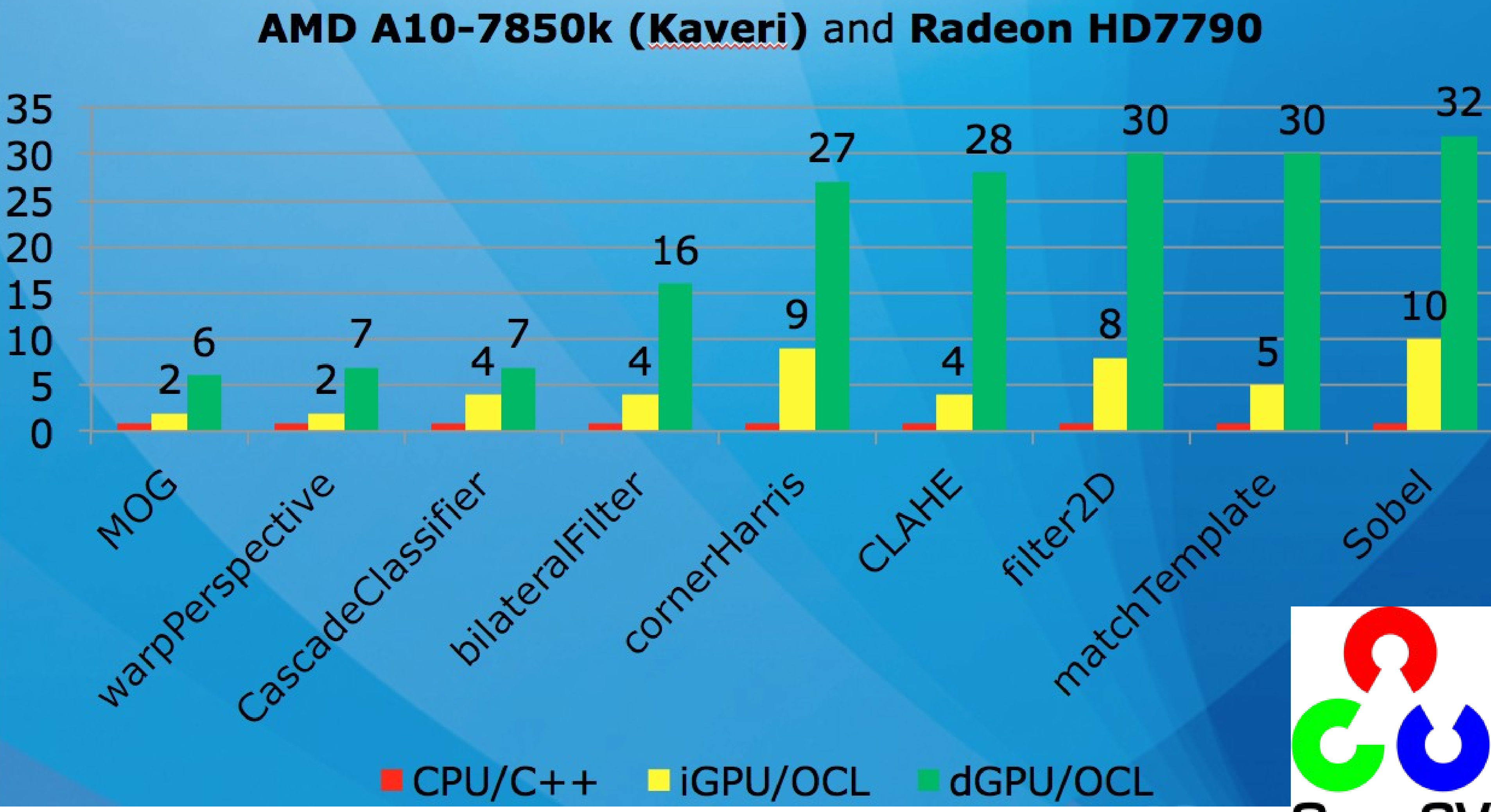


OpenCV and Hardware Acceleration

- OpenCV was a central basis for OpenVX
  - a hardware abstraction layer
  - for embedded vision acceleration
- OpenVX Supporters:



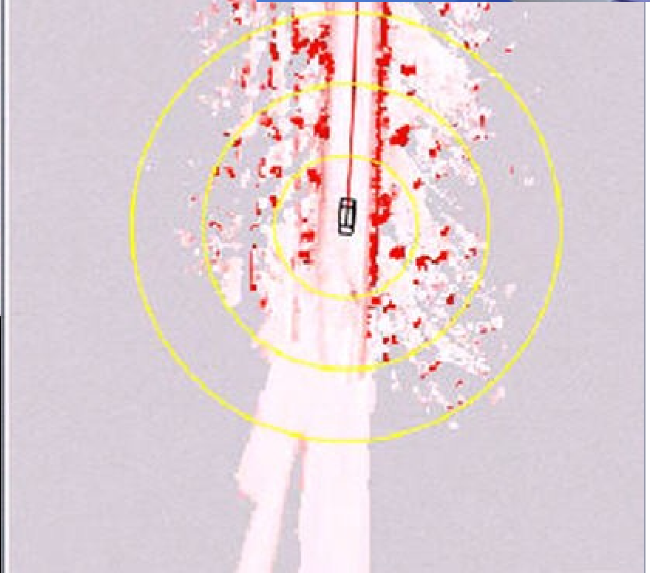
OpenCL™ performance in OpenCV 3.0



OpenCV

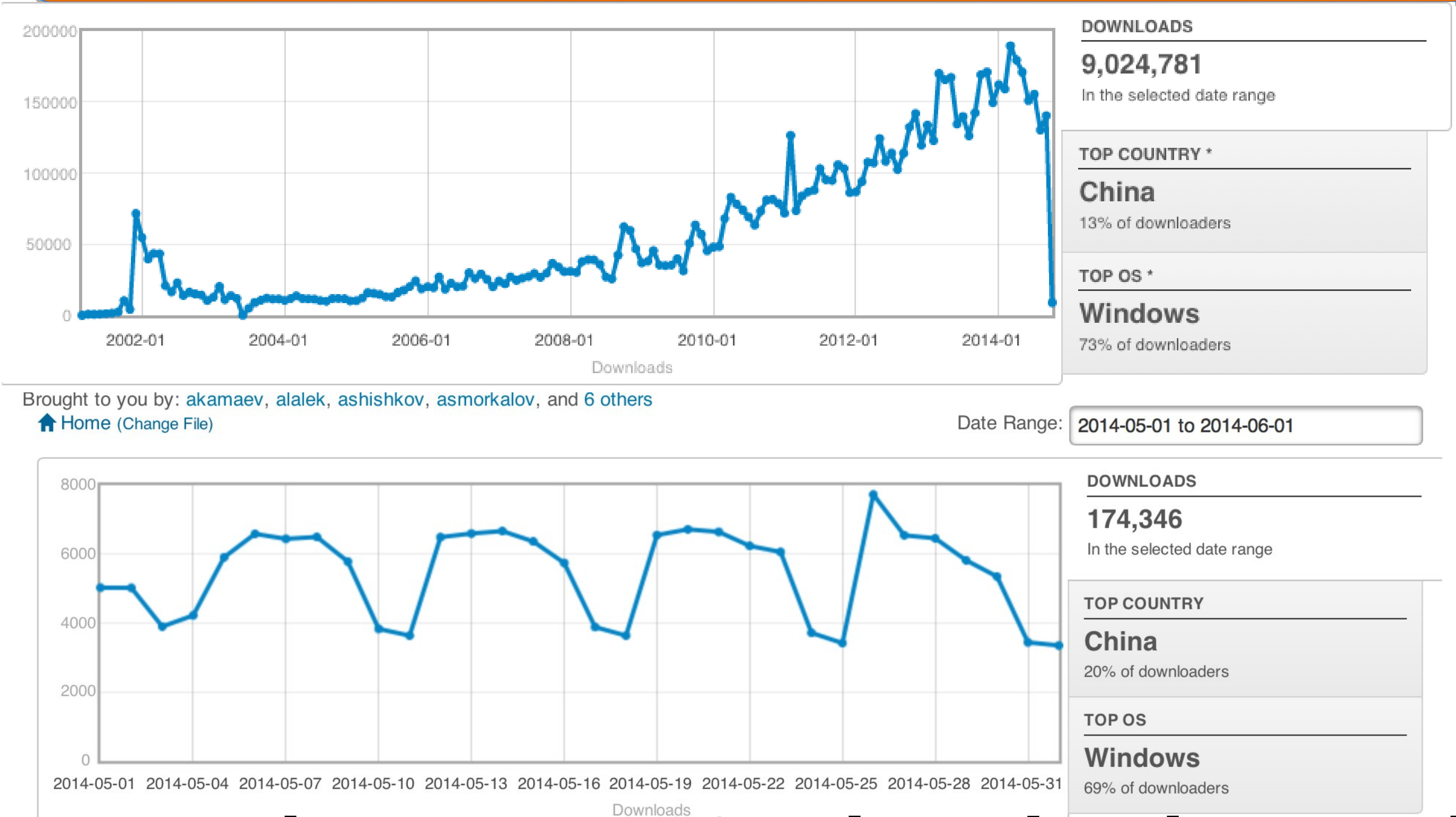
Where is OpenCV Used?

- 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, swimm
- Robotics – personal, industrial, hobby
- Coin production in China



Popularity

Over 9M downloads!



DOWNLOADS

9,024,781

In the selected date range

TOP COUNTRY *

China

13% of downloaders

TOP OS *

Windows

73% of downloaders

Brought to you by: akamaev, alalek, ashishkov, asmorkalov, and 6 others

Home (Change File)

Date Range: 2014-05-01 to 2014-06-01

DOWNLOADS

174,346

In the selected date range

TOP COUNTRY

China

20% of downloaders

TOP OS

Windows

69% of downloaders

Ramping to > 160K downloads/month

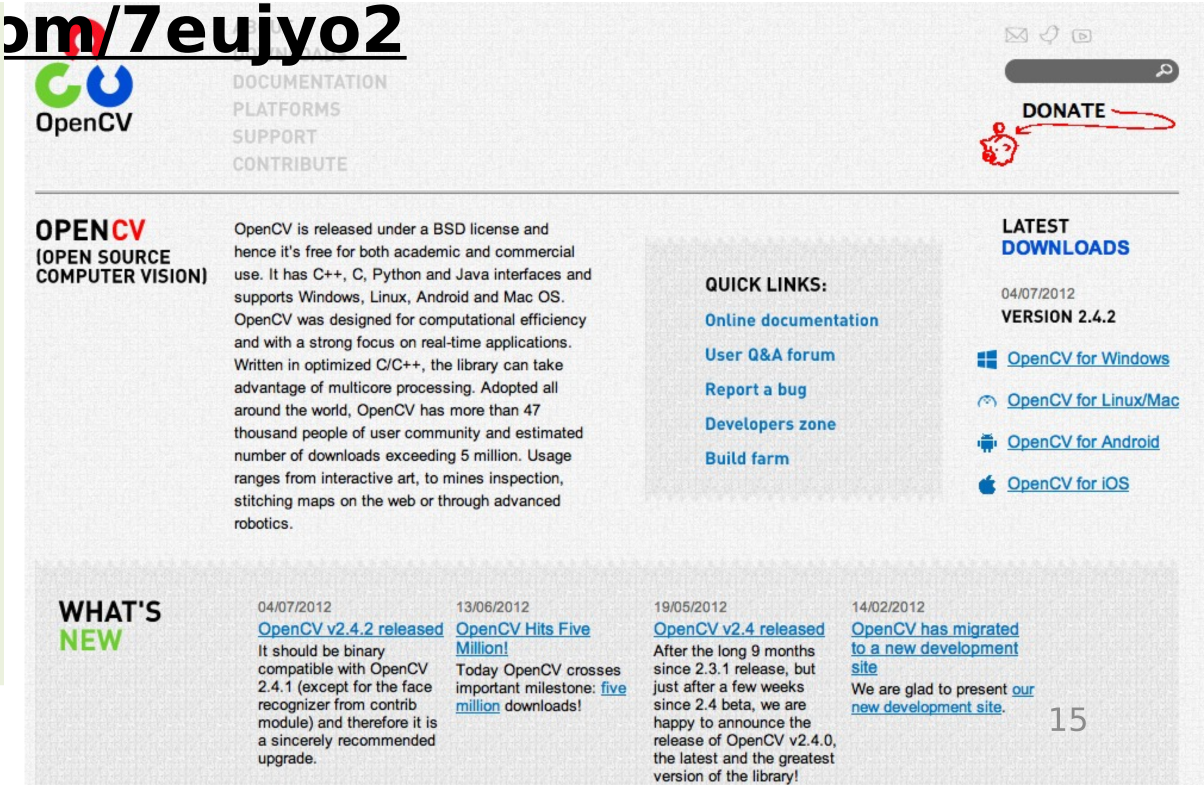
OpenCV Corporation

- Founded this July, 2012
- http://opencv.org (user site)
  Documentation:
  http://docs.opencv.org
- http://code.opencv.org (developer site)
- Contribute (via Credit, debit or paypal):
  om/7eujyo2

For corporate support
And/or partnership, contact
Garybradski@gmail.com

I am looking for entrepreneurial people to staff up OpenCV:

- Vision
- Business Dev
- Software
- Hardware



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

04/07/2012
OpenCV v2.4.2 released
It should be binary compatible with OpenCV 2.4.1 (except for the face recognizer from contrib module) and therefore it is a sincerely recommended upgrade.

13/06/2012
OpenCV Hits Five Million!
Today OpenCV crosses important milestone: five million downloads!

19/05/2012
OpenCV v2.4 released
After the long 9 months since 2.3.1 release, but just after a few weeks since 2.4 beta, we are happy to announce the release of OpenCV v2.4.0, the latest and the greatest version of the library!

14/02/2012
OpenCV has migrated to a new development site
We are glad to present our new development site.

What’s In OpenCV

- High level

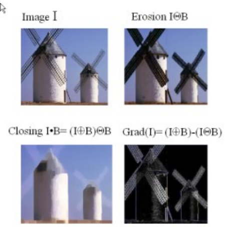
OpenCV Overview:

> 2500 algorithms

loper http://code.opencv.org; User: http://opencv.org

Robot suppo

General Image Processing Functions



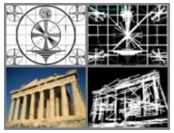
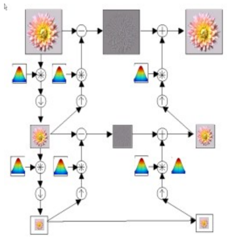
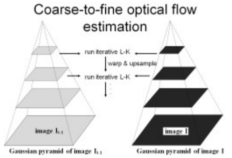
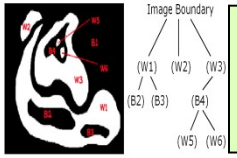


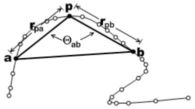
Image Pyramids

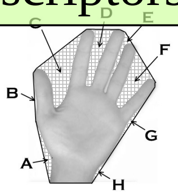



Coarse-to-fine optical flow estimation
run iterative L-K
warp & upsample
run iterative L-K
image I_{t-1}
image I
Gaussian pyramid of image I_{t-1}
Gaussian pyramid of image I

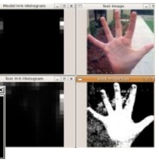
Geometric descriptors

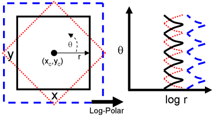




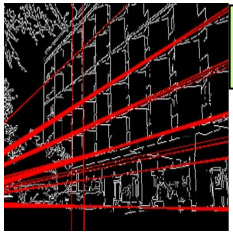


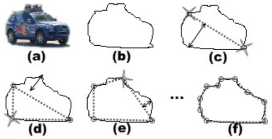
Segmentation



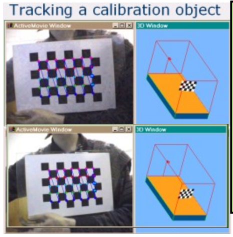


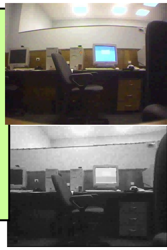
Features



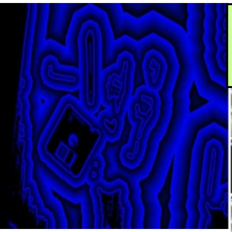


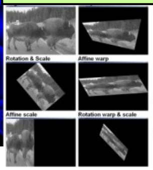
Camera calibration, Stereo, 3D

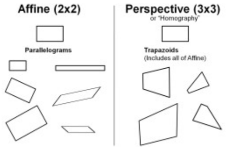




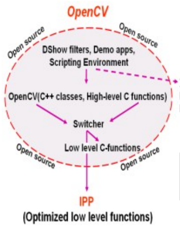
Transforms







Utilities and Data Structures


OpenCV

Open source

DShow filters, Demo apps, Scripting Environment

OpenCV(C++ classes, High-level C functions)

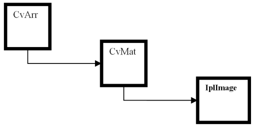
Intel Image Processing Library

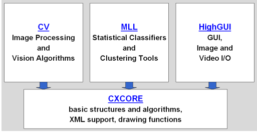
Switcher

Low level C-functions

IPP

(Optimized low level functions)



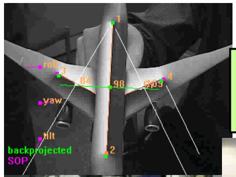

CV
Image Processing and Vision Algorithms

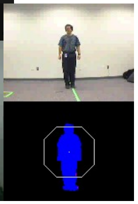
MLL
Statistical Classifiers and Clustering Tools

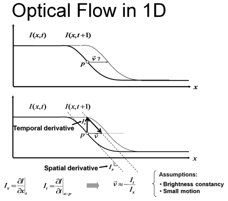
HighGUI
GUI, Image and Video I/O

CXCORE
basic structures and algorithms, XML support, drawing functions

Tracking

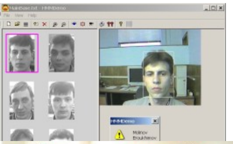


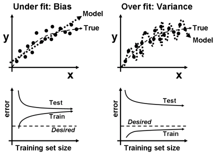




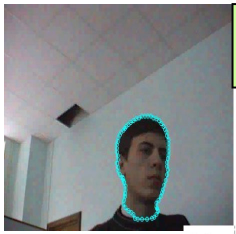
Machine Learning:

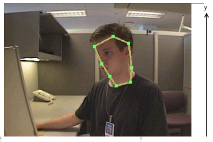
- Detection,
- Recognition

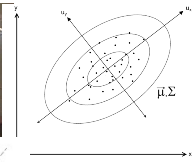


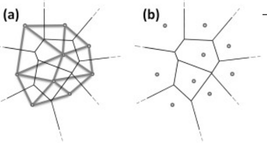


Fitting

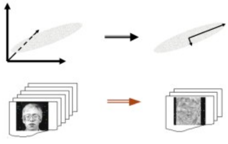








Matrix Math



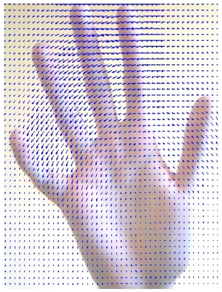
OpenCV Algorithm Modules Overview

HighGUI:
I/O, Interface

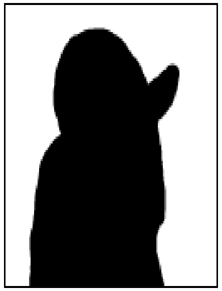
Image
Processing

Transforms

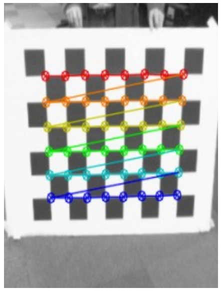
Fitting



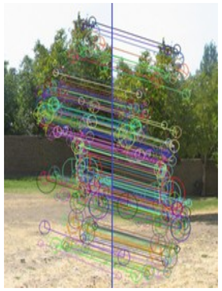
Optical Flow
Tracking



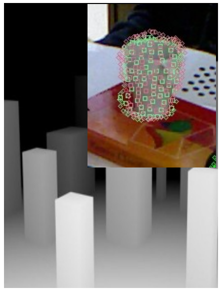
Segmentatio
n



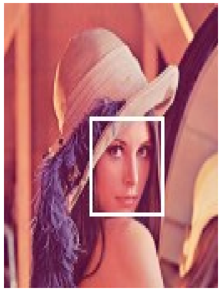
Calibration



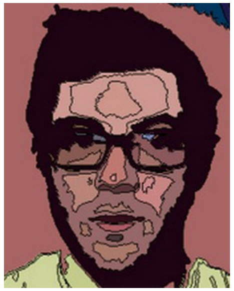
Features
VSLAM



Depth, Pose
Normals,
Planes, 3D
Features



Object
recognition
Machine



Computationa
l Photography

CORE:
Data structures, Matrix math, Exceptions
etc

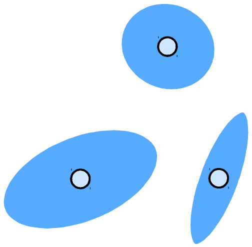
Machine Learning Library (MLL)

CLASSIFICATION / REGRESSION

Fast Approximate NN (FLANN)
Extremely Random Trees
CART
Naïve Bayes
MLP (Back propagation)
Statistical Boosting, 4 flavors
Random Forests
SVM
Face Detector
(Histogram matching)
(Correlation)

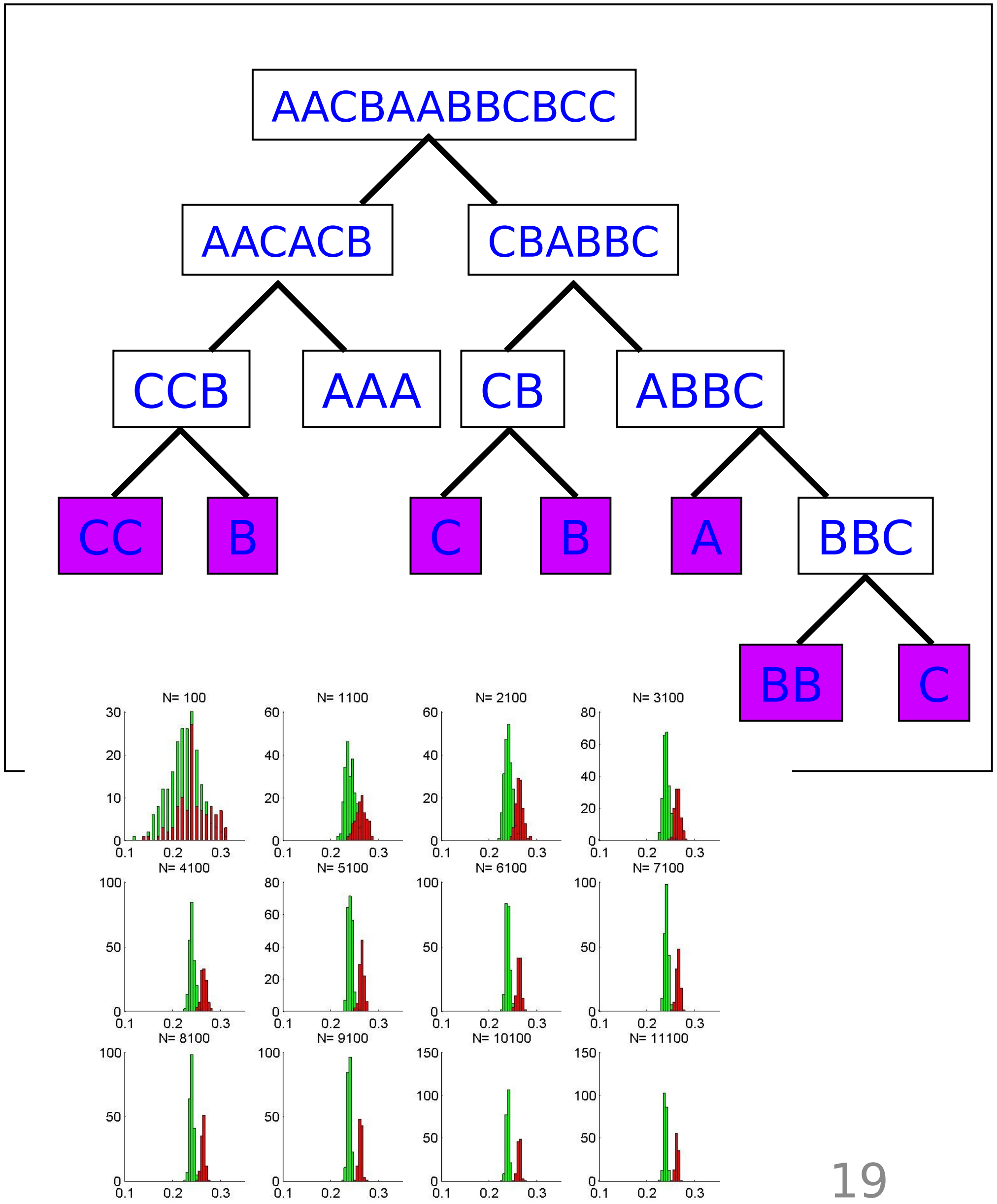
CLUSTERING

K-Means
EM
(Mahalanobis distance)



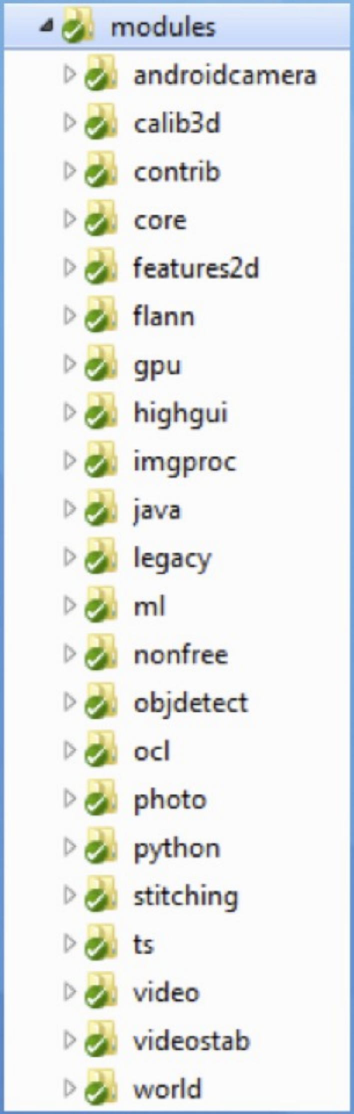
TUNING/VALIDATION

Cross validation
Bootstrapping
Variable importance
Sampling methods



http://opencv.org

Modules



modules
- androidcamera
- calib3d
- contrib
- core
- features2d
- flann
- gpu
- highgui
- imgproc
- java
- legacy
- ml
- nonfree
- objdetect
- ocl
- photo
- python
- stitching
- ts
- video
- videostab
- world

Algorithmic

- **core, imgproc, calib3d, video, ml, objdetect, features2d**
- **photo, stitching, videostab, superres**
- **contrib, legacy, nonfree, flann**

GPU

- **gpu, ocl**

Infrastructure

- **highgui, world**
- **python, java**
- **ts, androidcamera**

C vs C++ API: Focus Detector

C

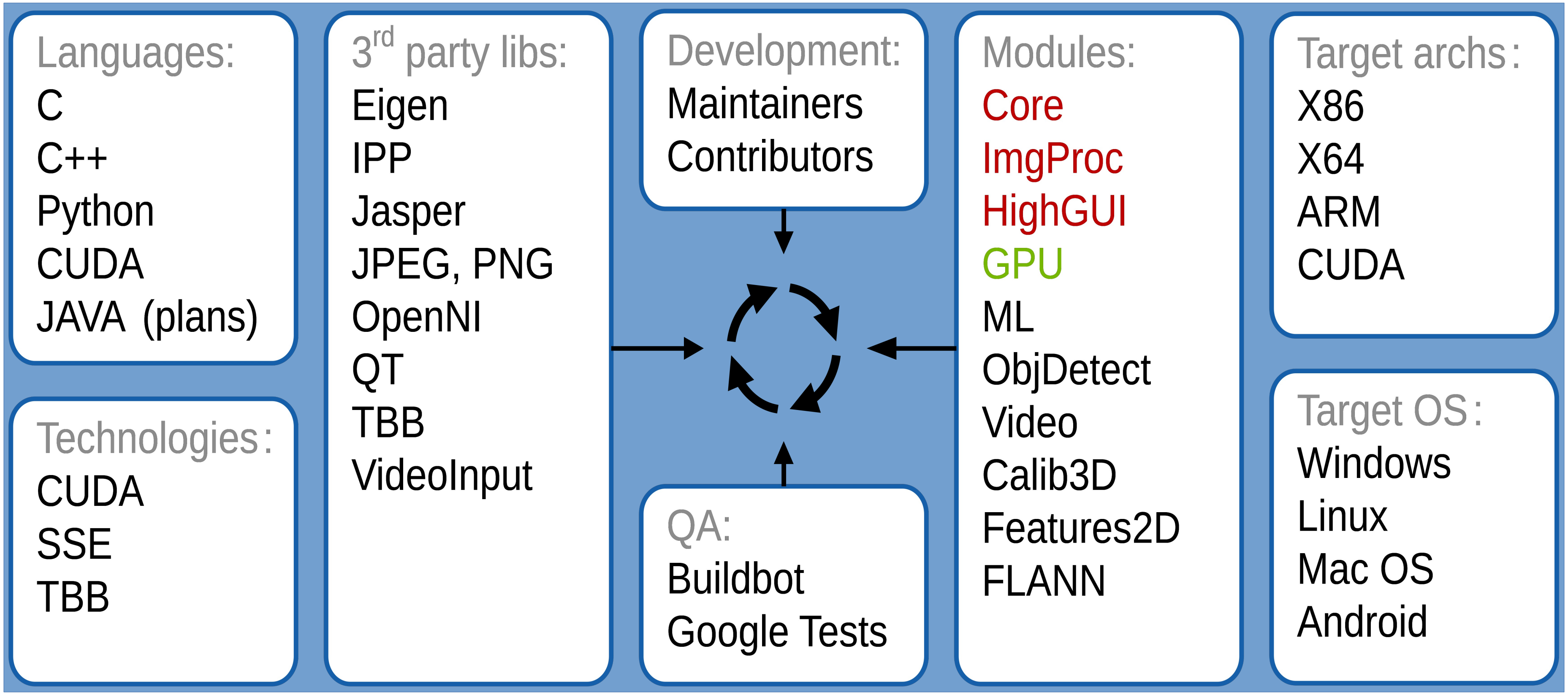
```
double calcGradients(const IplImage *src,
                     int aperture_size = 7)
{
  CvSize sz = cvGetSize(src);

  IplImage* img16_x = cvCreateImage(sz, IPL_DEPTH_16S, 1);
  IplImage* img16_y = cvCreateImage(sz, IPL_DEPTH_16S, 1);
  cvSobel(src, img16_x, 1, 0, aperture_size);
  cvSobel(src, img16_y, 0, 1, aperture_size);

  IplImage* imgF_x = cvCreateImage(sz, IPL_DEPTH_32F, 1);
  IplImage* imgF_y = cvCreateImage(sz, IPL_DEPTH_32F, 1);
  cvScale(img16_x, imgF_x);
  cvScale(img16_y, imgF_y);

  IplImage* magnitude = cvCreateImage(sz, IPL_DEPTH_32F, 1);
  cvCartToPolar(imgF_x, imgF_y, magnitude);
  double res = cvSum(magnitude).val[0];

  cvReleaseImage(&magnitude );
  cvReleaseImage(&imgF_x);
  cvReleaseImage(&imgF_y);
  cvReleaseImage(&img16_x);
  cvReleaseImage(&img16_y);

  return res;
}
```

C++

```
double contrast_measure(Mat& img)
{
  Mat dx, dy;

  Sobel(img, dx, 1, 0, 3, CV_32F);
  Sobel(img, dy, 0, 1, 3, CV_32F);
  magnitude(dx, dy, dx);

  return sum(dx)[0];
}
```

itseez

OpenCV Architecture and Development



Languages:
C
C++
Python
CUDA
JAVA (plans)

Technologies:
CUDA
SSE
TBB

3rd party libs:
Eigen
IPP
Jasper
JPEG, PNG
OpenNI
QT
TBB
VideoInput

Development:
Maintainers
Contributors

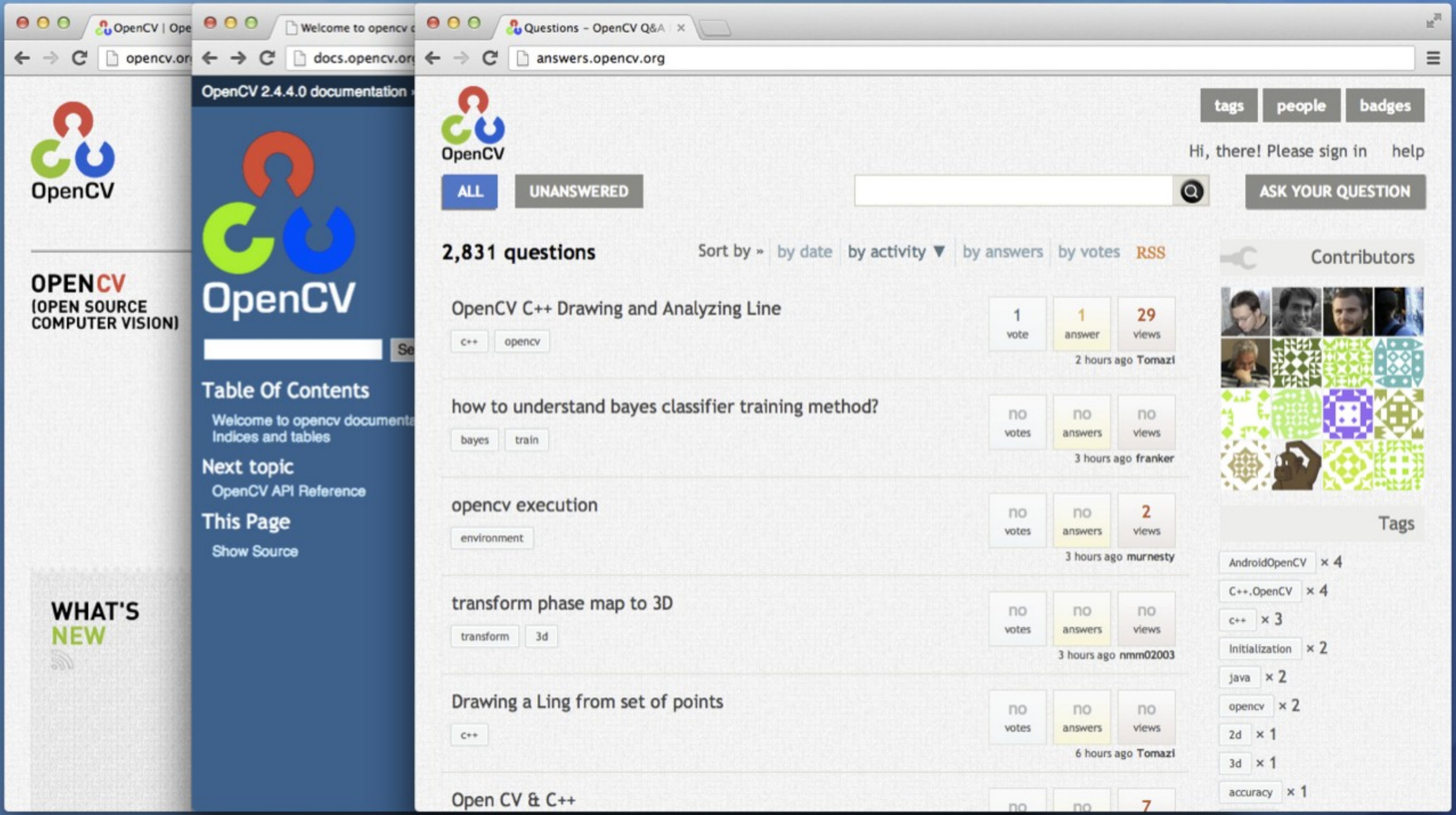
QA:
Buildbot
Google Tests

Modules:
Core
ImgProc
HighGUI
GPU
ML
ObjDetect
Video
Calib3D
Features2D
FLANN

Target archs:
X86
X64
ARM
CUDA

Target OS:
Windows
Linux
Mac OS
Android

Web resources



2,831 questions

OpenCV C++ Drawing and Analyzing Line — c++, opencv — 1 vote, 1 answer, 29 views — 2 hours ago Tomazi

how to understand bayes classifier training method? — bayes, train — no votes, no answers, no views — 3 hours ago franker

opencv execution — environment — no votes, no answers, 2 views — 3 hours ago murnesty

transform phase map to 3D — transform, 3d — no votes, no answers, no views — 3 hours ago nmm02003

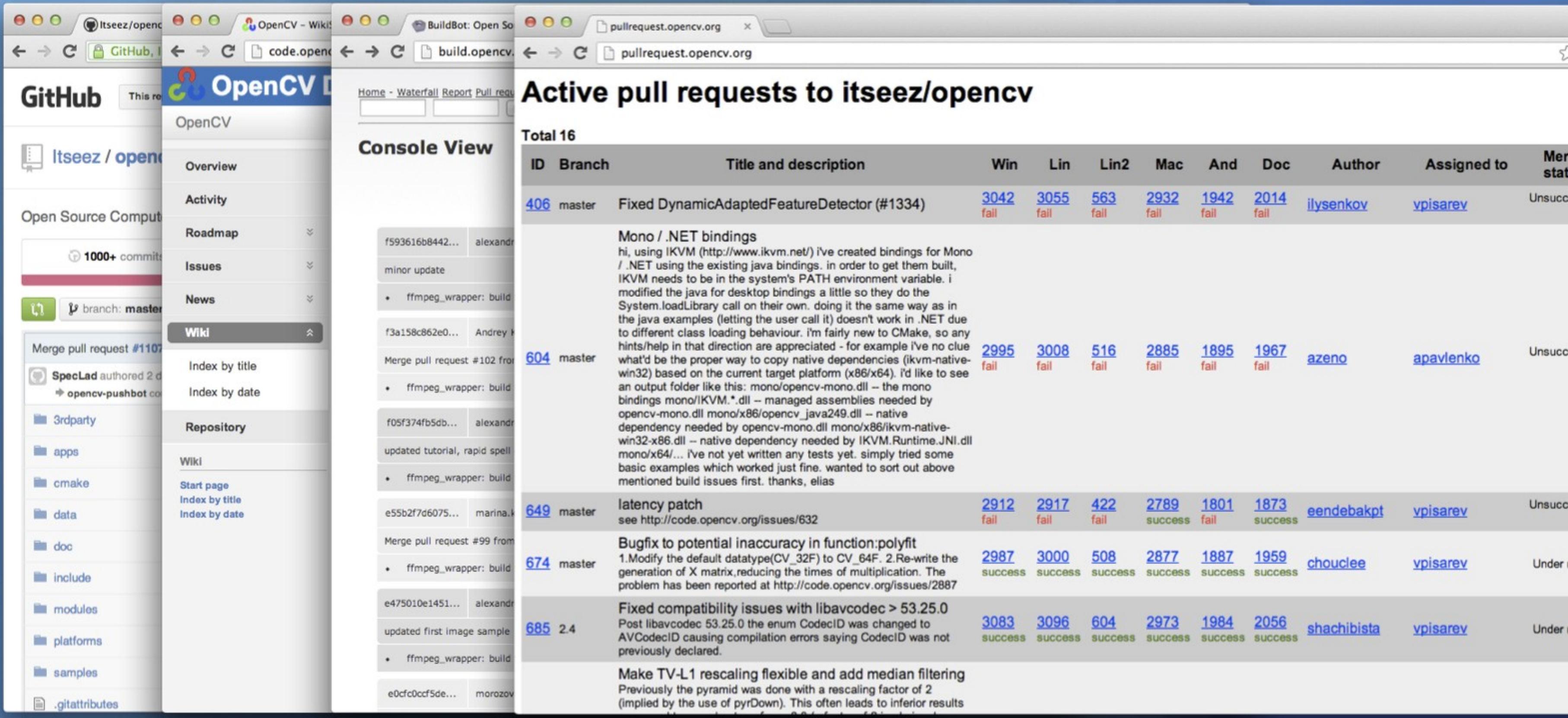
Drawing a Ling from set of points — c++ — no votes, no answers, no views — 6 hours ago Tomazi

Open CV & C++ — no, no, 7

Tags	
AndroidOpenCV	× 4
C++.OpenCV	× 4
c++	× 3
initialization	× 2
java	× 2
opencv	× 2
2d	× 1
3d	× 1
accuracy	× 1

opencv.org, docs.opencv.org, answers.opencv.org

Development infrastructure



Console View

```
f593616b8442...   alexandr
minor update
•   ffmpeg_wrapper: build
f3a158c862e0...   Andrey K
Merge pull request #102 fro
•   ffmpeg_wrapper: build
f05f374fb5db...   alexandr
updated tutorial, rapid spell
•   ffmpeg_wrapper: build
e55b2f7d6075...   marina.k
Merge pull request #99 from
•   ffmpeg_wrapper: build
e475010e1451...   alexandr
updated first image sample
•   ffmpeg_wrapper: build
e0cfc0ccf5de...   morozov
```

Active pull requests to itseez/opencv

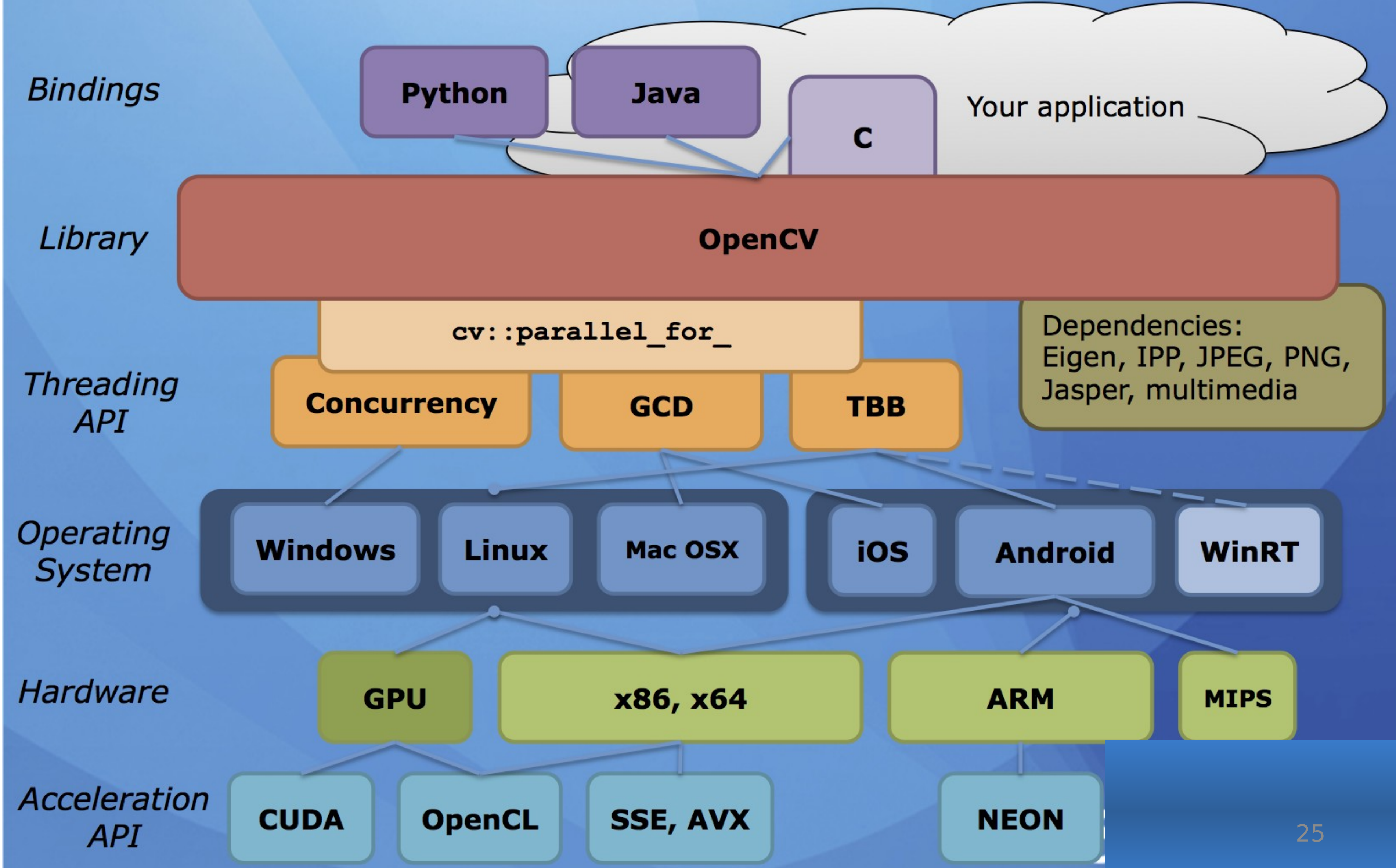
Total 16

ID	Branch	Title and description	Win	Lin	Lin2	Mac	And	Doc	Author	Assigned to	Mer stat
406	master	Fixed DynamicAdaptedFeatureDetector (#1334)	3042 fail	3055 fail	563 fail	2932 fail	1942 fail	2014 fail	ilysenkov	vpisarev	Unsucc
604	master	Mono / .NET bindings hi, using IKVM (http://www.ikvm.net/) i've created bindings for Mono / .NET using the existing java bindings. in order to get them built, IKVM needs to be in the system's PATH environment variable. i modified the java for desktop bindings a little so they do the System.loadLibrary call on their own. doing it the same way as in the java examples (letting the user call it) doesn't work in .NET due to different class loading behaviour. i'm fairly new to CMake, so any hints/help in that direction are appreciated - for example i've no clue what'd be the proper way to copy native dependencies (ikvm-native-win32) based on the current target platform (x86/x64). i'd like to see an output folder like this: mono/opencv-mono.dll -- the mono bindings mono/IKVM.*.dll -- managed assemblies needed by opencv-mono.dll mono/x86/opencv_java249.dll -- native dependency needed by opencv-mono.dll mono/x86/ikvm-native-win32-x86.dll -- native dependency needed by IKVM.Runtime.JNI.dll mono/x64/... i've not yet written any tests yet. simply tried some basic examples which worked just fine. wanted to sort out above mentioned build issues first. thanks, elias	2995 fail	3008 fail	516 fail	2885 fail	1895 fail	1967 fail	azeno	apavlenko	Unsucc
649	master	latency patch see http://code.opencv.org/issues/632	2912 fail	2917 fail	422 fail	2789 success	1801 fail	1873 success	eendebakpt	vpisarev	Unsucc
674	master	Bugfix to potential inaccuracy in function:polyfit 1.Modify the default datatype(CV_32F) to CV_64F. 2.Re-write the generation of X matrix,reducing the times of multiplication. The problem has been reported at http://code.opencv.org/issues/2887	2987 success	3000 success	508 success	2877 success	1887 success	1959 success	chouclee	vpisarev	Under
685	2.4	Fixed compatibility issues with libavcodec > 53.25.0 Post libavcodec 53.25.0 the enum CodecID was changed to AVCodecID causing compilation errors saying CodecID was not previously declared.	3083 success	3096 success	604 success	2973 success	1984 success	2056 success	shachibista	vpisarev	Under
		Make TV-L1 rescaling flexible and add median filtering Previously the pyramid was done with a rescaling factor of 2 (implied by the use of pyrDown). This often leads to inferior results									

code.opencv.org, build.opencv.org,

https://github.com/itseez/opencv

OpenCV Environment



Bindings

Python

Java

C

Your application

Library

OpenCV

Threading API

cv::parallel_for_

Concurrency

GCD

TBB

Dependencies:
Eigen, IPP, JPEG, PNG,
Jasper, multimedia

Operating System

Windows

Linux

Mac OSX

iOS

Android

WinRT

Hardware

GPU

x86, x64

ARM

MIPS

Acceleration API

CUDA

OpenCL

SSE, AVX

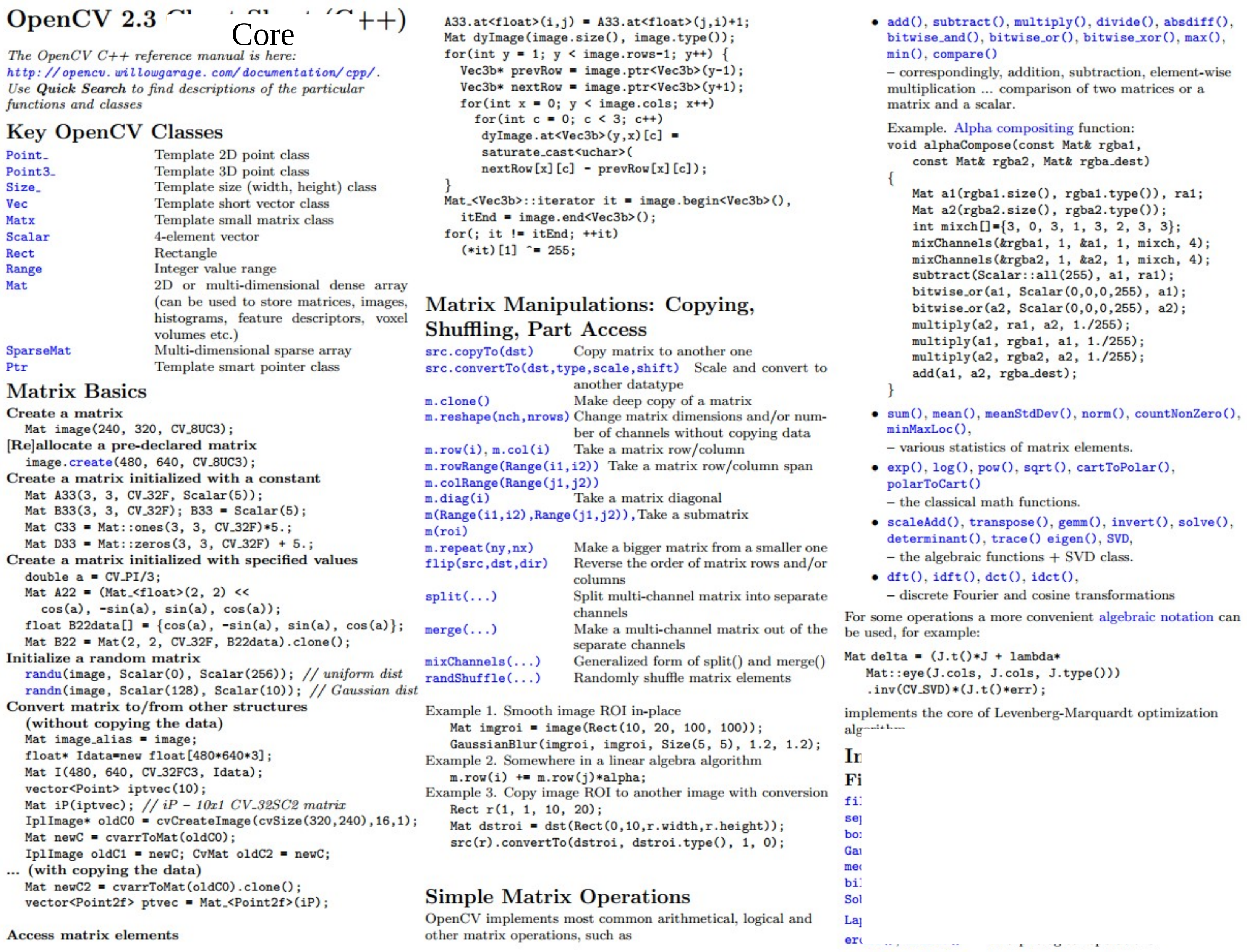
NEON

What’s In OpenCV 3.0

- Modules

CORE:
Data structures,
Matrix math,
Exceptions etc.

OpenCV Modules: Core



Core

OpenCV 2.3 C ++)

The OpenCV C++ reference manual is here: *http://opencv.willowgarage.com/documentation/cpp/.* *Use* ***Quick Search*** *to find descriptions of the particular functions and classes*

Key OpenCV Classes

Point_	Template 2D point class
Point3_	Template 3D point class
Size_	Template size (width, height) class
Vec	Template short vector class
Matx	Template small matrix class
Scalar	4-element vector
Rect	Rectangle
Range	Integer value range
Mat	2D or multi-dimensional dense array (can be used to store matrices, images, histograms, feature descriptors, voxel volumes etc.)
SparseMat	Multi-dimensional sparse array
Ptr	Template smart pointer class

Matrix Basics

Create a matrix

```
   Mat image(240, 320, CV_8UC3);
```

[Re]allocate a pre-declared matrix

```
   image.create(480, 640, CV_8UC3);
```

Create a matrix initialized with a constant

```
   Mat A33(3, 3, CV_32F, Scalar(5));
   Mat B33(3, 3, CV_32F); B33 = Scalar(5);
   Mat C33 = Mat::ones(3, 3, CV_32F)*5.;
   Mat D33 = Mat::zeros(3, 3, CV_32F) + 5.;
```

Create a matrix initialized with specified values

```
   double a = CV_PI/3;
   Mat A22 = (Mat_<float>(2, 2) <<
     cos(a), -sin(a), sin(a), cos(a));
   float B22data[] = {cos(a), -sin(a), sin(a), cos(a)};
   Mat B22 = Mat(2, 2, CV_32F, B22data).clone();
```

Initialize a random matrix

```
   randu(image, Scalar(0), Scalar(256)); // uniform dist
   randn(image, Scalar(128), Scalar(10)); // Gaussian dist
```

Convert matrix to/from other structures
(without copying the data)

```
   Mat image_alias = image;
   float* Idata=new float[480*640*3];
   Mat I(480, 640, CV_32FC3, Idata);
   vector<Point> iptvec(10);
   Mat iP(iptvec); // iP – 10x1 CV_32SC2 matrix
   IplImage* oldC0 = cvCreateImage(cvSize(320,240),16,1);
   Mat newC = cvarrToMat(oldC0);
   IplImage oldC1 = newC; CvMat oldC2 = newC;
```

... (with copying the data)

```
   Mat newC2 = cvarrToMat(oldC0).clone();
   vector<Point2f> ptvec = Mat_<Point2f>(iP);
```

Access matrix elements

```
A33.at<float>(i,j) = A33.at<float>(j,i)+1;
Mat dyImage(image.size(), image.type());
for(int y = 1; y < image.rows-1; y++) {
  Vec3b* prevRow = image.ptr<Vec3b>(y-1);
  Vec3b* nextRow = image.ptr<Vec3b>(y+1);
  for(int x = 0; y < image.cols; x++)
    for(int c = 0; c < 3; c++)
     dyImage.at<Vec3b>(y,x)[c] =
     saturate_cast<uchar>(
     nextRow[x][c] - prevRow[x][c]);
}
Mat_<Vec3b>::iterator it = image.begin<Vec3b>(),
  itEnd = image.end<Vec3b>();
for(; it != itEnd; ++it)
  (*it)[1] ^= 255;
```

Matrix Manipulations: Copying, Shuffling, Part Access

src.copyTo(dst)	Copy matrix to another one
src.convertTo(dst,type,scale,shift)	Scale and convert to another datatype
m.clone()	Make deep copy of a matrix
m.reshape(nch,nrows)	Change matrix dimensions and/or number of channels without copying data
m.row(i), m.col(i)	Take a matrix row/column
m.rowRange(Range(i1,i2)) m.colRange(Range(j1,j2))	Take a matrix row/column span
m.diag(i)	Take a matrix diagonal
m(Range(i1,i2),Range(j1,j2)), m(roi)	Take a submatrix
m.repeat(ny,nx)	Make a bigger matrix from a smaller one
flip(src,dst,dir)	Reverse the order of matrix rows and/or columns
split(...)	Split multi-channel matrix into separate channels
merge(...)	Make a multi-channel matrix out of the separate channels
mixChannels(...)	Generalized form of split() and merge()
randShuffle(...)	Randomly shuffle matrix elements

Example 1. Smooth image ROI in-place

```
   Mat imgroi = image(Rect(10, 20, 100, 100));
   GaussianBlur(imgroi, imgroi, Size(5, 5), 1.2, 1.2);
```

Example 2. Somewhere in a linear algebra algorithm

```
   m.row(i) += m.row(j)*alpha;
```

Example 3. Copy image ROI to another image with conversion

```
   Rect r(1, 1, 10, 20);
   Mat dstroi = dst(Rect(0,10,r.width,r.height));
   src(r).convertTo(dstroi, dstroi.type(), 1, 0);
```

Simple Matrix Operations

OpenCV implements most common arithmetical, logical and other matrix operations, such as

- add(), subtract(), multiply(), divide(), absdiff(), bitwise_and(), bitwise_or(), bitwise_xor(), max(), min(), compare()

  – correspondingly, addition, subtraction, element-wise multiplication ... comparison of two matrices or a matrix and a scalar.

  Example. Alpha compositing function:

  ```
  void alphaCompose(const Mat& rgba1,
     const Mat& rgba2, Mat& rgba_dest)
  {
     Mat a1(rgba1.size(), rgba1.type()), ra1;
     Mat a2(rgba2.size(), rgba2.type());
     int mixch[]={3, 0, 3, 1, 3, 2, 3, 3};
     mixChannels(&rgba1, 1, &a1, 1, mixch, 4);
     mixChannels(&rgba2, 1, &a2, 1, mixch, 4);
     subtract(Scalar::all(255), a1, ra1);
     bitwise_or(a1, Scalar(0,0,0,255), a1);
     bitwise_or(a2, Scalar(0,0,0,255), a2);
     multiply(a2, ra1, a2, 1./255);
     multiply(a1, rgba1, a1, 1./255);
     multiply(a2, rgba2, a2, 1./255);
     add(a1, a2, rgba_dest);
  }
  ```

- sum(), mean(), meanStdDev(), norm(), countNonZero(), minMaxLoc(),

  – various statistics of matrix elements.

- exp(), log(), pow(), sqrt(), cartToPolar(), polarToCart()

  – the classical math functions.

- scaleAdd(), transpose(), gemm(), invert(), solve(), determinant(), trace() eigen(), SVD,

  – the algebraic functions + SVD class.

- dft(), idft(), dct(), idct(),

  – discrete Fourier and cosine transformations

For some operations a more convenient algebraic notation can be used, for example:

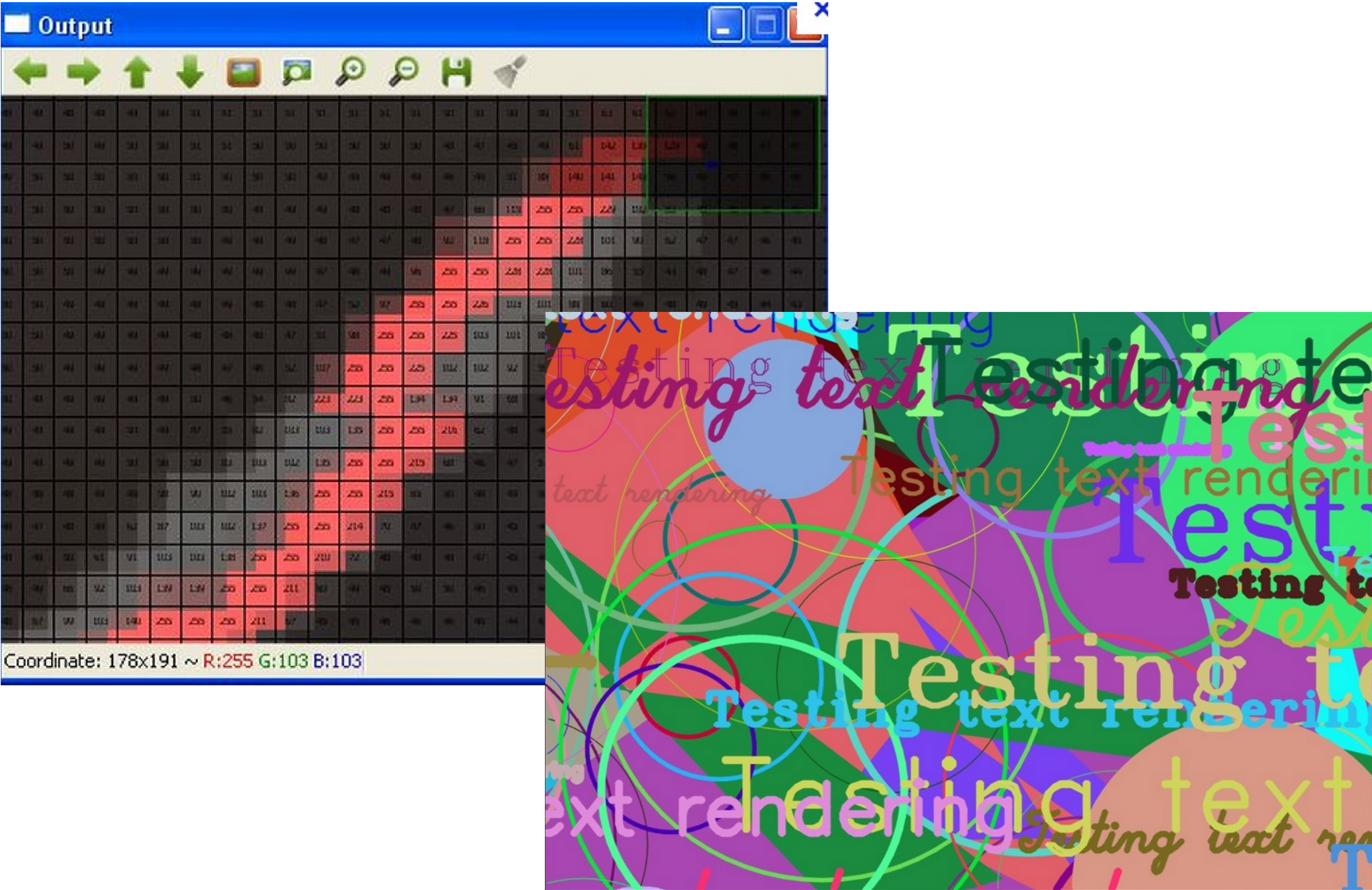
```
Mat delta = (J.t()*J + lambda*
   Mat::eye(J.cols, J.cols, J.type()))
   .inv(CV_SVD)*(J.t()*err);
```

implements the core of Levenberg-Marquardt optimization alg

In
Fi
fi
se
bo
Ga
me
bi
Sol
La
er

HighGUI:
I/O, Interface

penCV Modules: HighGUI

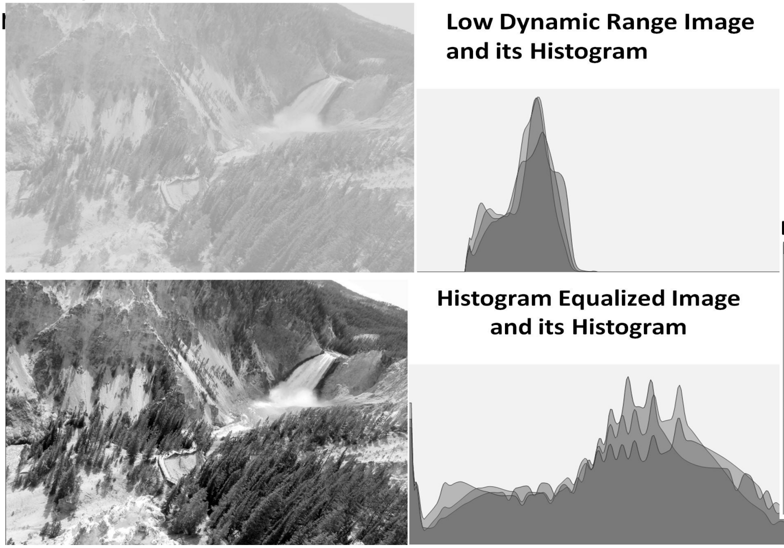


Coordinate: 178x191 ~ R:255 G:103 B:103

OpenCV Modules: Image Processing

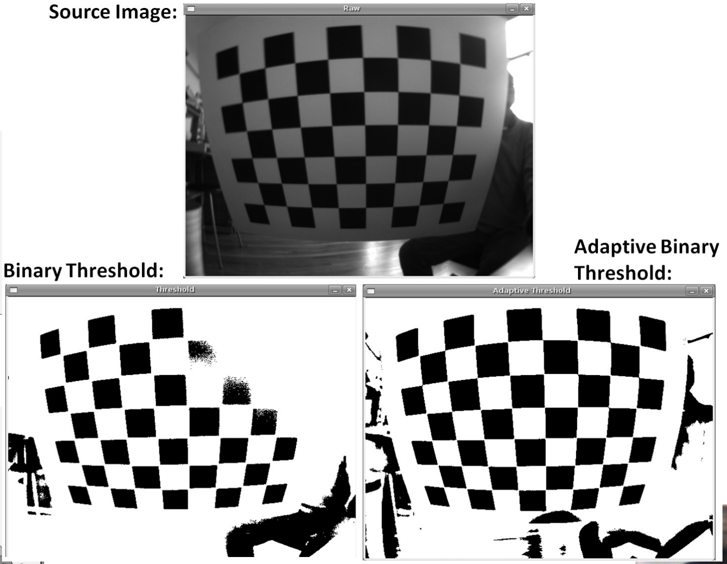
Image

Pr



Low Dynamic Range Image and its Histogram

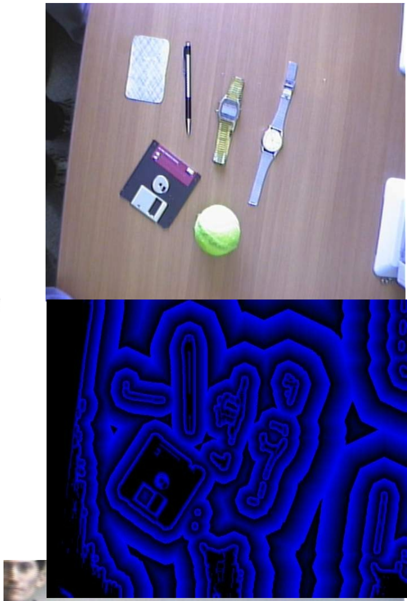
Histogram Equalized Image and its Histogram

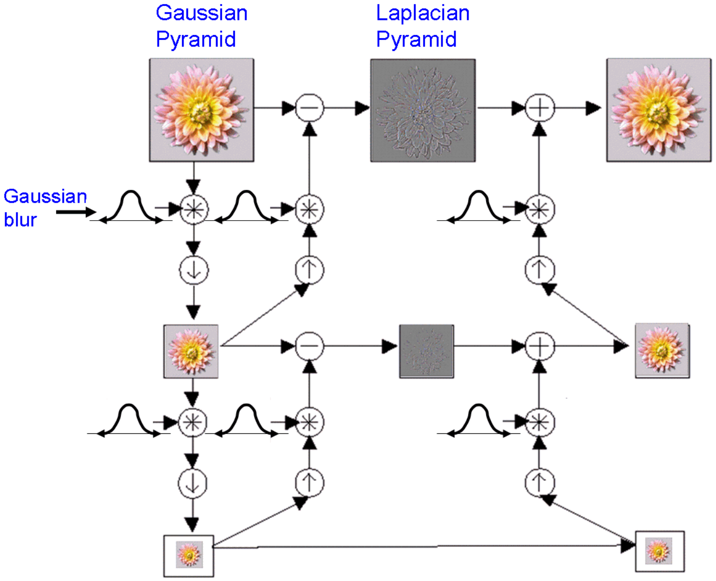


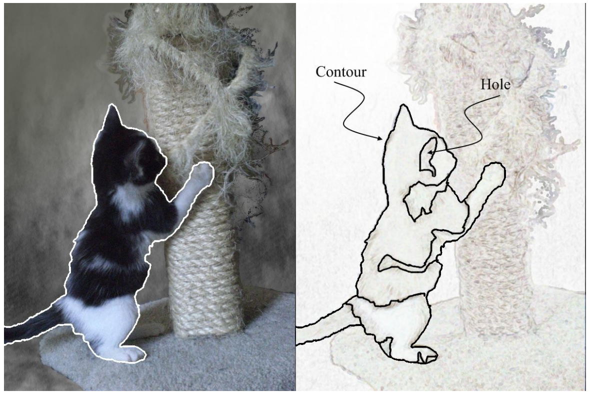
Source Image:

Binary Threshold:

Adaptive Binary Threshold:

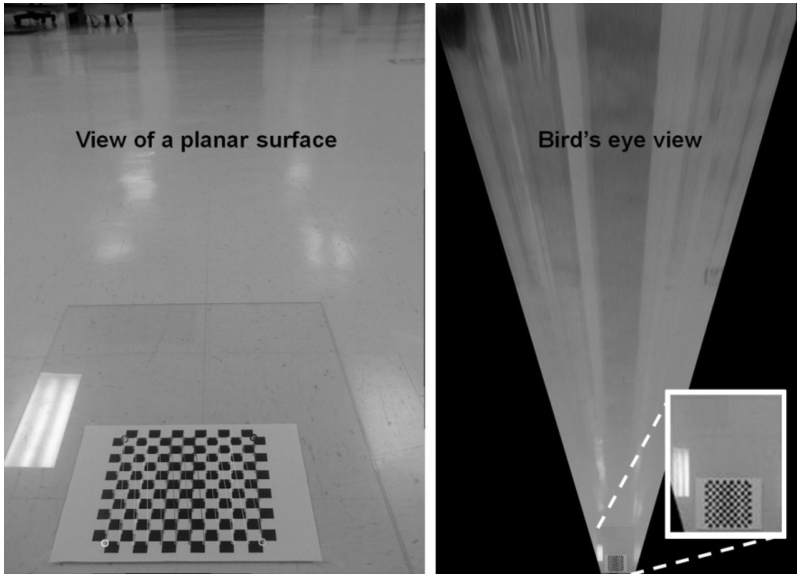






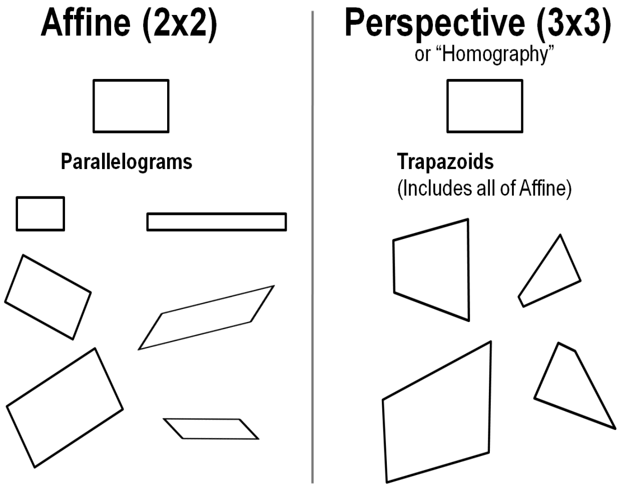
OpenCV Modules: Transforms

Transfor



View of a planar surface

Bird's eye view



Affine (2x2)

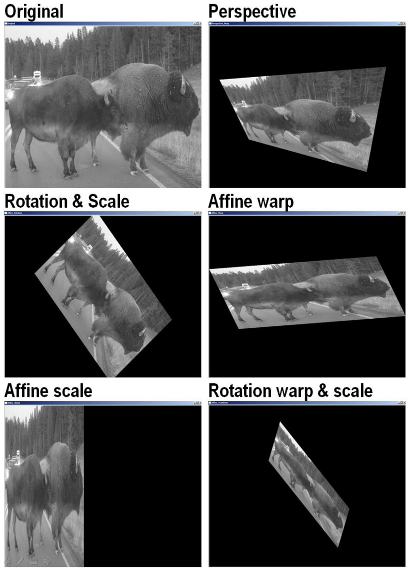
Parallelograms

Perspective (3x3)

or "Homography"

Trapazoids

(Includes all of Affine)



Original

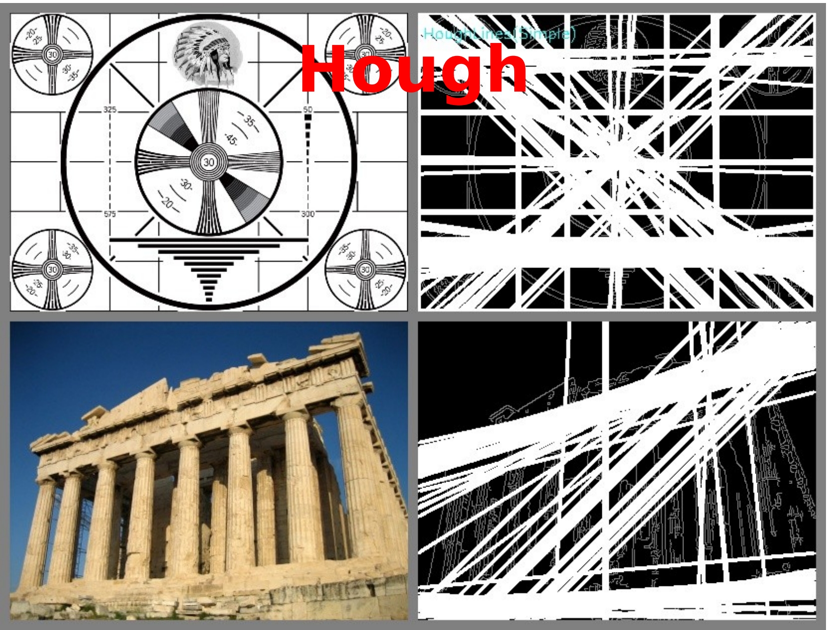
Perspective

Rotation & Scale

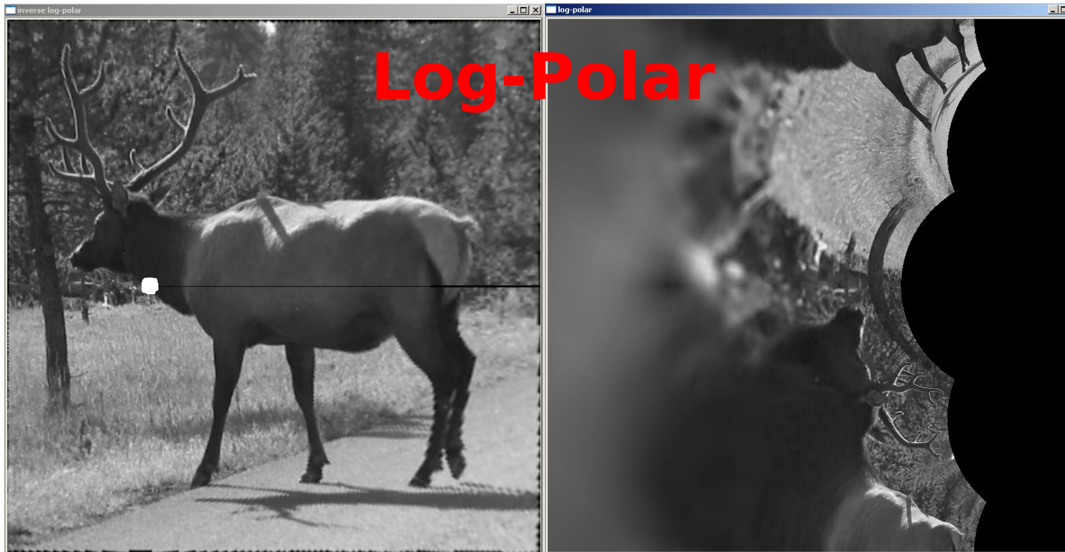
Affine warp

Affine scale

Rotation warp & scale

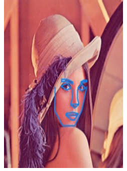


Hough

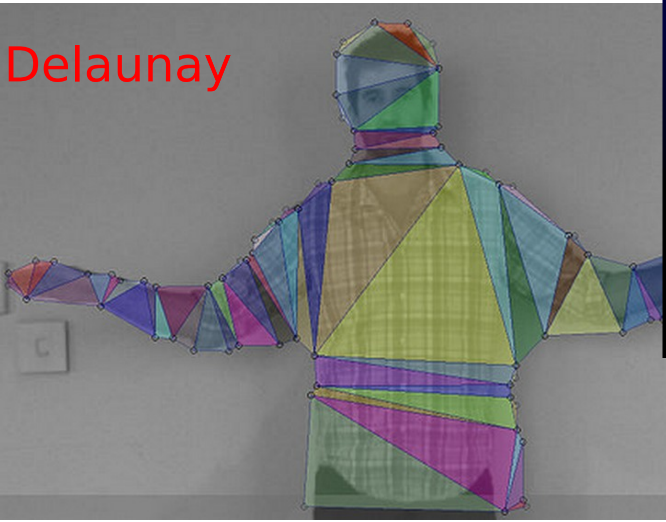


Log-Polar

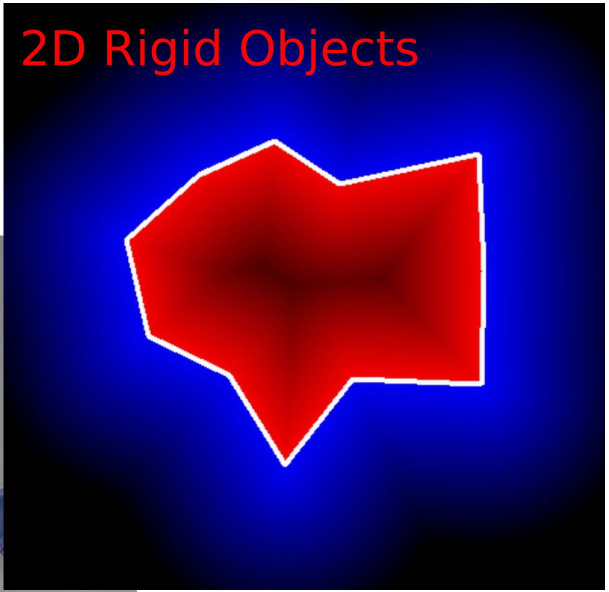
OpenCV Modules: Fitting



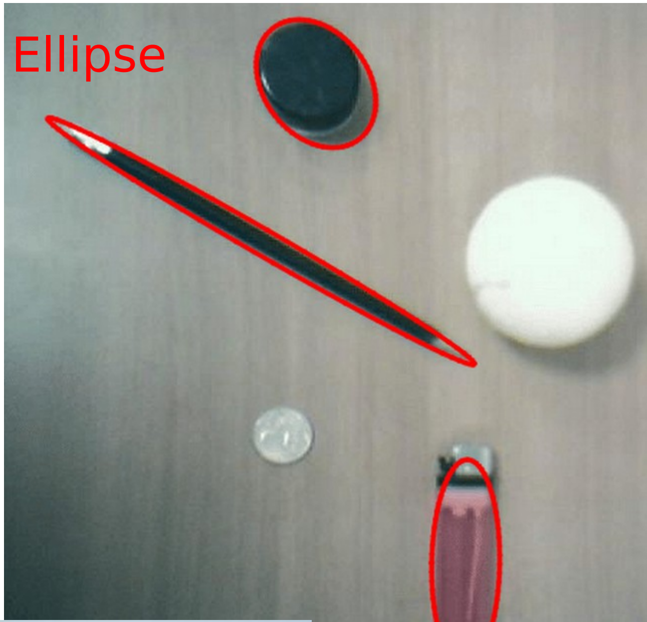
Fitting



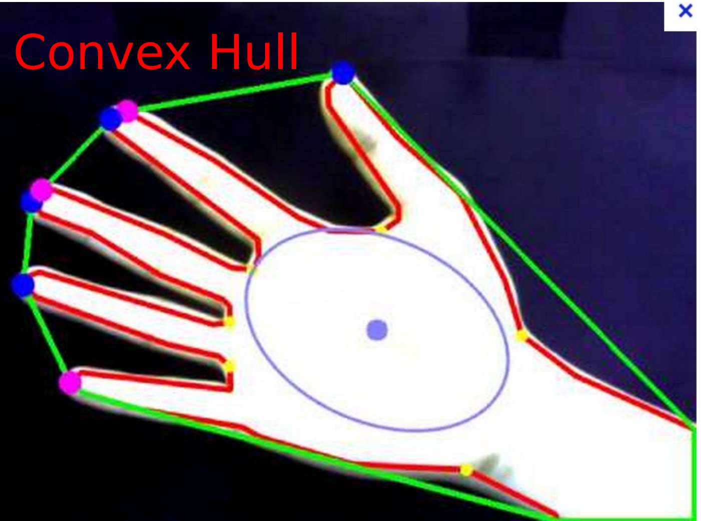
Delaunay



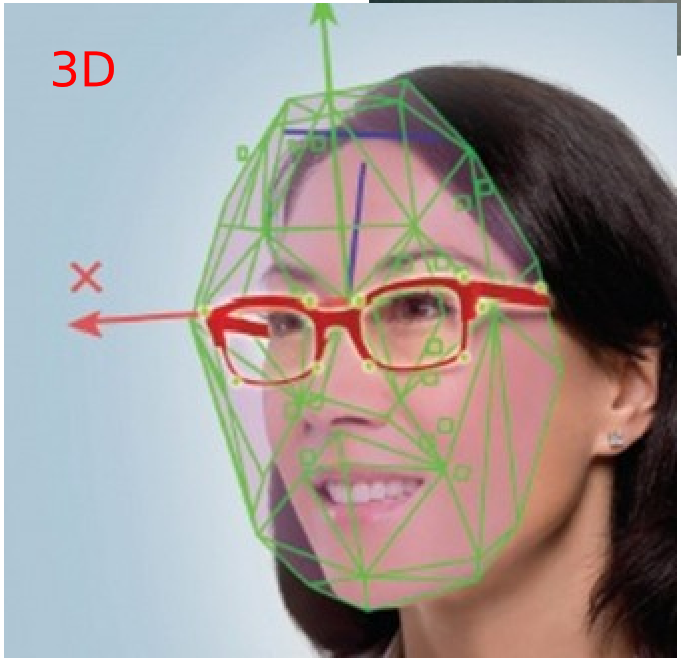
2D Rigid Objects



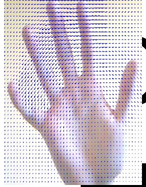
Ellipse



Convex Hull



3D



Optical Flow Tracking

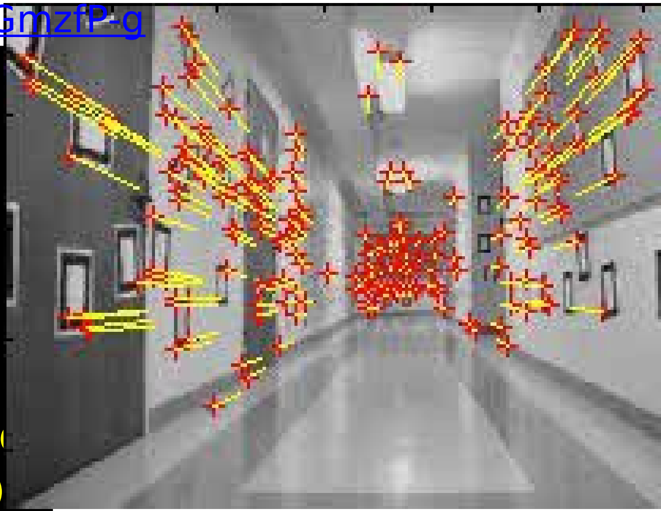
OpenCV Modules: Optic Flow, Track

http://www.youtube.com/watch?v=bWyBGmzfP-g

```
// opencv/samples/c/lkdemo.c
int main(…){
…
CvCapture* capture = <…> ?
      cvCaptureFromCAM(camera_id) :
      cvCaptureFromFile(path);
if( !capture ) return -1;
for(;;) {
   IplImage* frame=cvQueryFrame(capture);
   if(!frame) break;
   // … copy and process image
cvCalcOpticalFlowPyrLK( …)
   cvShowImage( "LkDemo", result );
   c=cvWaitKey(30); // run at ~20-30fps speed
   if(c >= 0) {
       // process key
 }}
cvReleaseCapture(&capture
```

lkdemo.c, 190 lin

(needs camera to run)

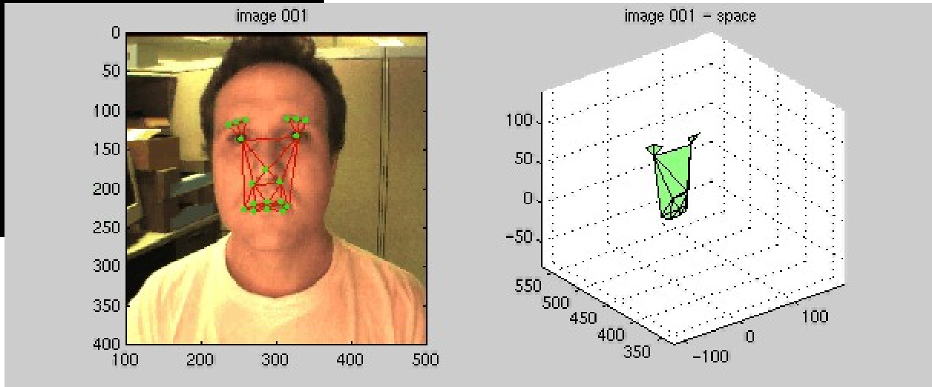


$$I(x+dx, y+dy, t+dt) = I(x,y,t);$$

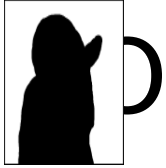
$$-\partial I/\partial t = \partial I/\partial x \cdot (dx/dt) + \partial I/\partial y \cdot (dy/dt);$$

$$G \cdot \partial X = b,$$

$$\partial X = (\partial x, \partial y), G = \sum \begin{bmatrix} I_x^2, & I_x I_y \\ I_x I_y, & I_y^2 \end{bmatrix}, b = \sum I_t \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

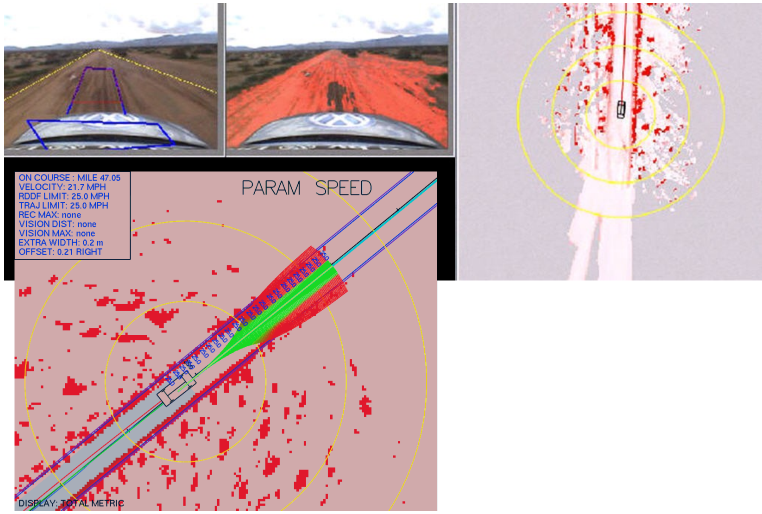


http://www.youtube.com/watch?v=1osj7kRgswk

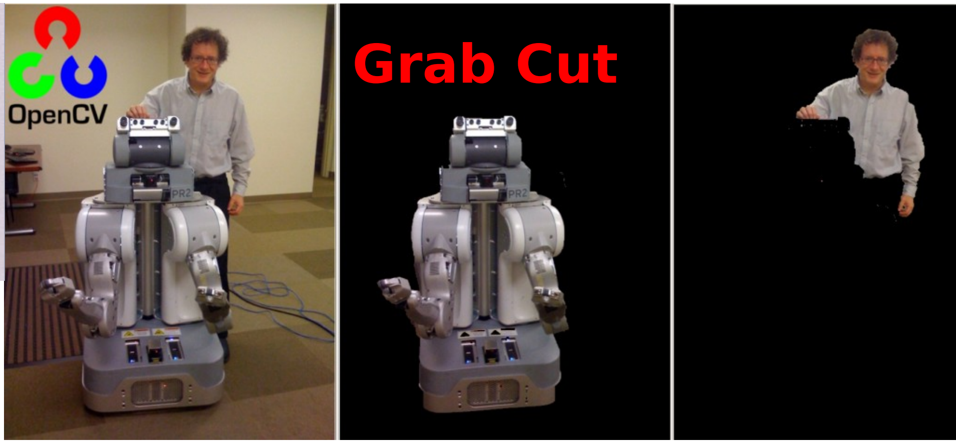


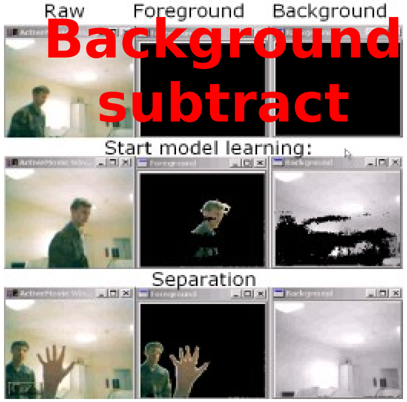
Segmentati on

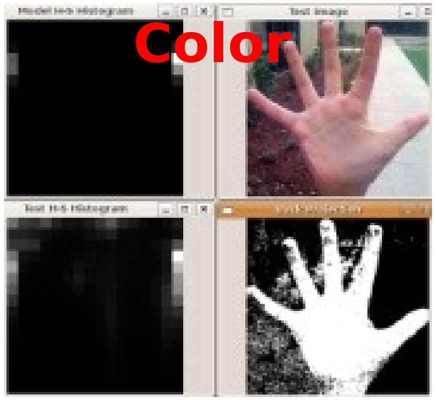
OpenCV Modules: Segmentation



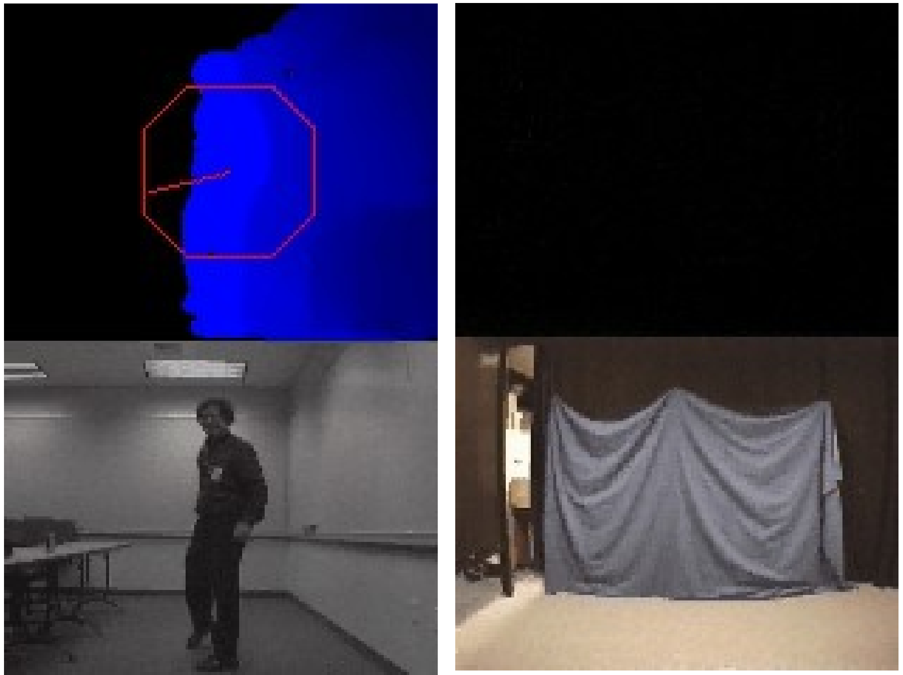
```
ON COURSE : MILE 47.05
VELOCITY: 21.7 MPH
RDDF LIMIT: 25.0 MPH
TRAJ LIMIT: 25.0 MPH
REC MAX: none
VISION DIST: none
VISION MAX: none
EXTRA WIDTH: 0.2 m
OFFSET: 0.21 RIGHT
```

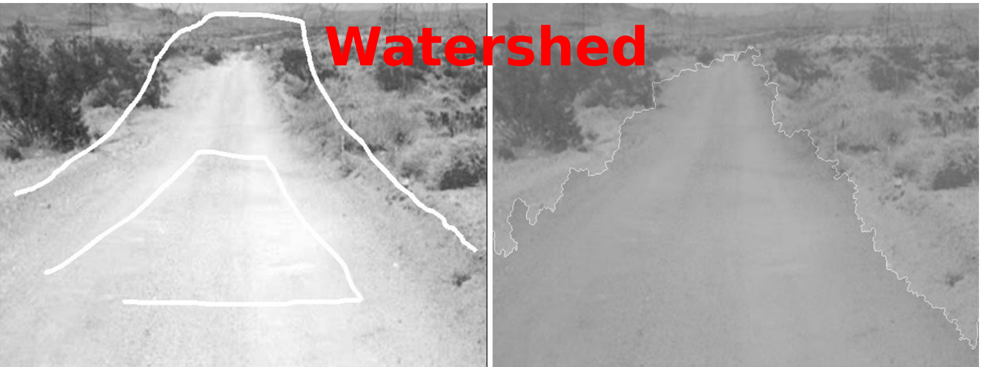


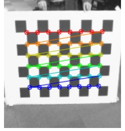




https://www.youtube.com/watch?v=OxmDonZja
http://www.youtube.com/watch?v=Ktrjh5-KLKo



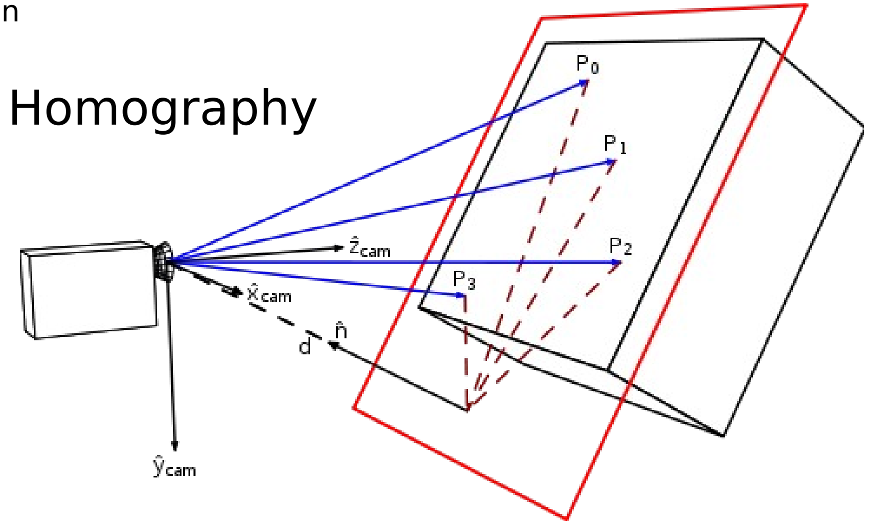


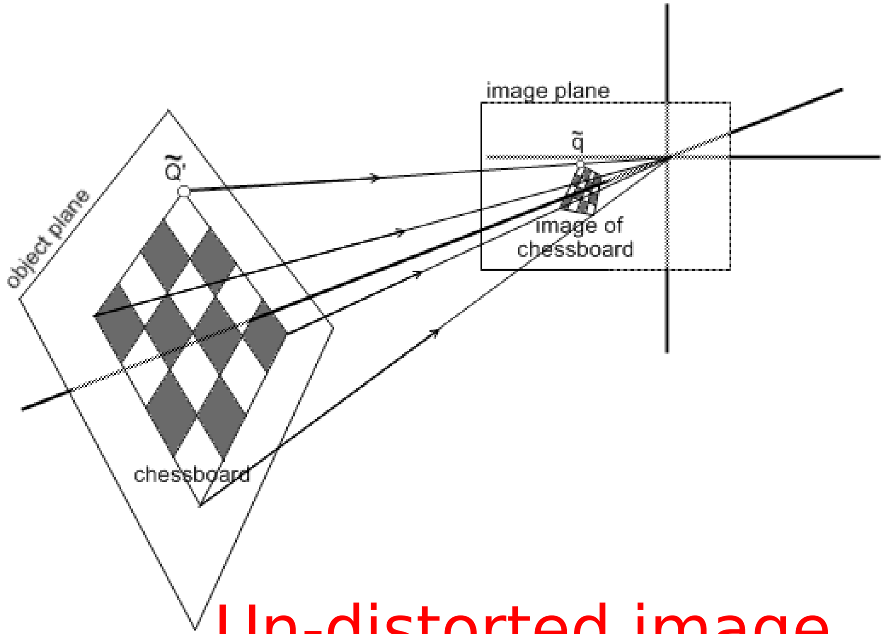


Calibrati
on

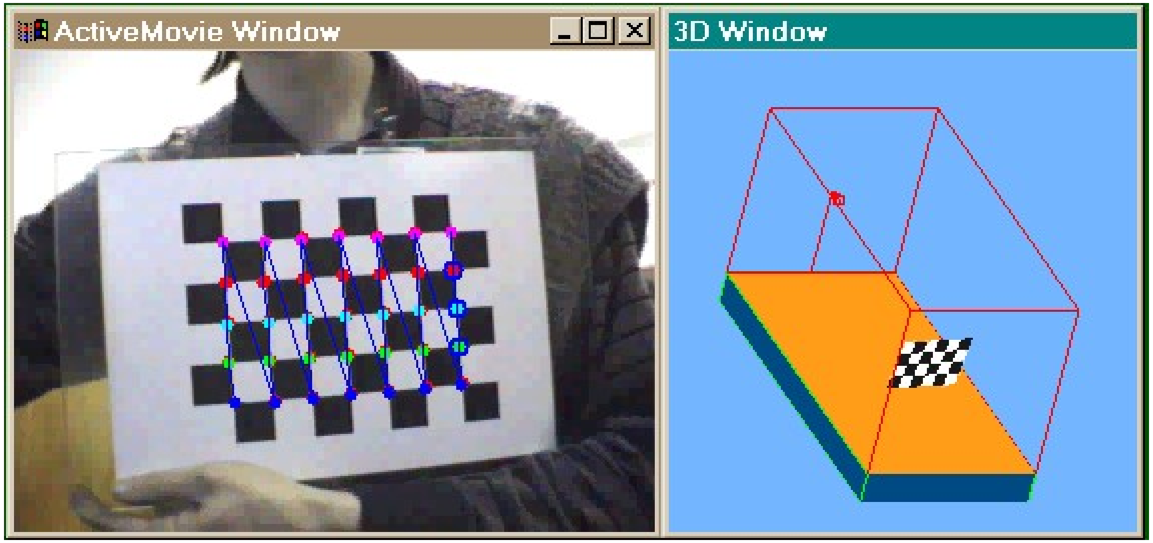
OpenCV Modules: Calibration

Homography





3D view of checkerboard

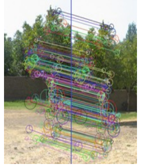


Un-distorted image

http://www.youtube.com/watch?v=DrXIQfQHFv0

http://www.youtube.com/watch?v=PuWQ

OpenCV Modules: Features, VSLAM



Features
VSLAM

Read two input images:

```
Mat img1 = imread(argv[1], CV_LOAD_IMAGE_GRAYSCAL
```

Detect keypoints in both images:

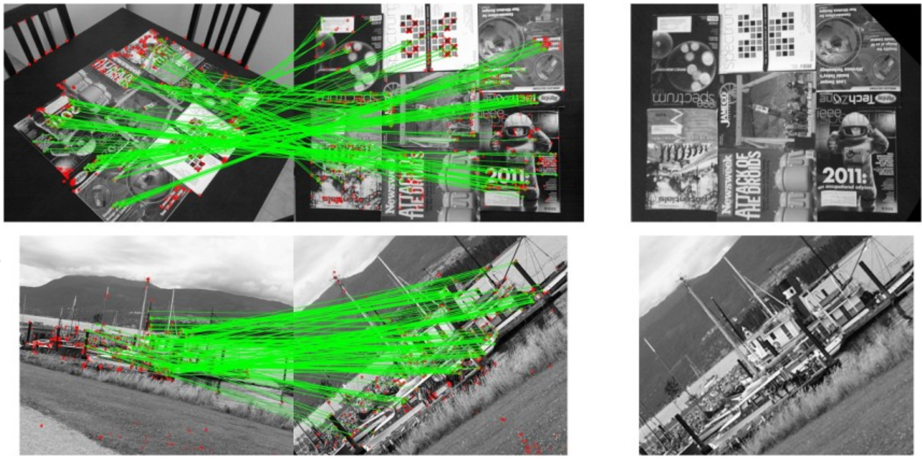
```
// detecting keypoints
FastFeatureDetector detector(15);
vector<KeyPoint> keypoints1;
detector.detect(img1, keypoints1);
```

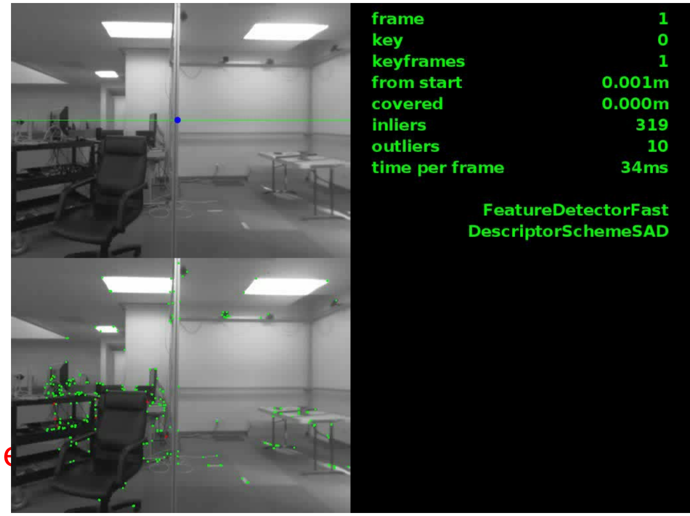
Compute descriptors for each of the keypoints:

```
// computing descriptors
SurfDescriptorExtractor extractor;
Mat descriptors1;
extractor.compute(img1, keypoints1, descriptors1);
```

Now, find the closest matches between descriptors from the first image

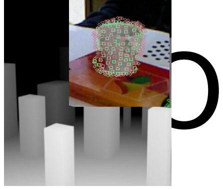
```
// matching descriptors
BruteForceMatcher<L2<float> > matcher;
vector<DMatch> matches;
matcher.match(descriptors1, descriptors2, matches);
```





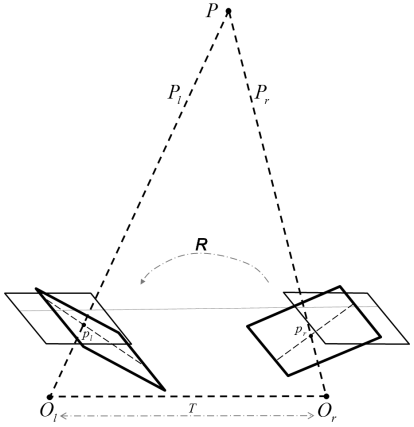
```
frame                1
key                  0
keyframes            1
from start      0.001m
covered         0.000m
inliers            319
outliers            10
time per frame    34ms

      FeatureDetectorFast
   DescriptorSchemeSAD
```

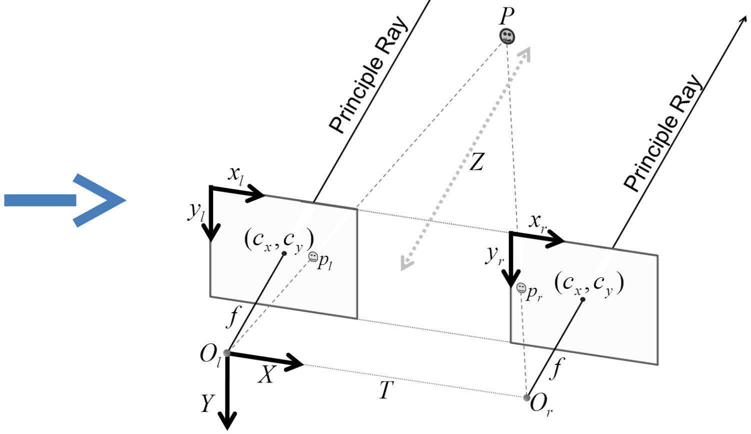
Change one or both of these lines to switch detector and/or descriptor types

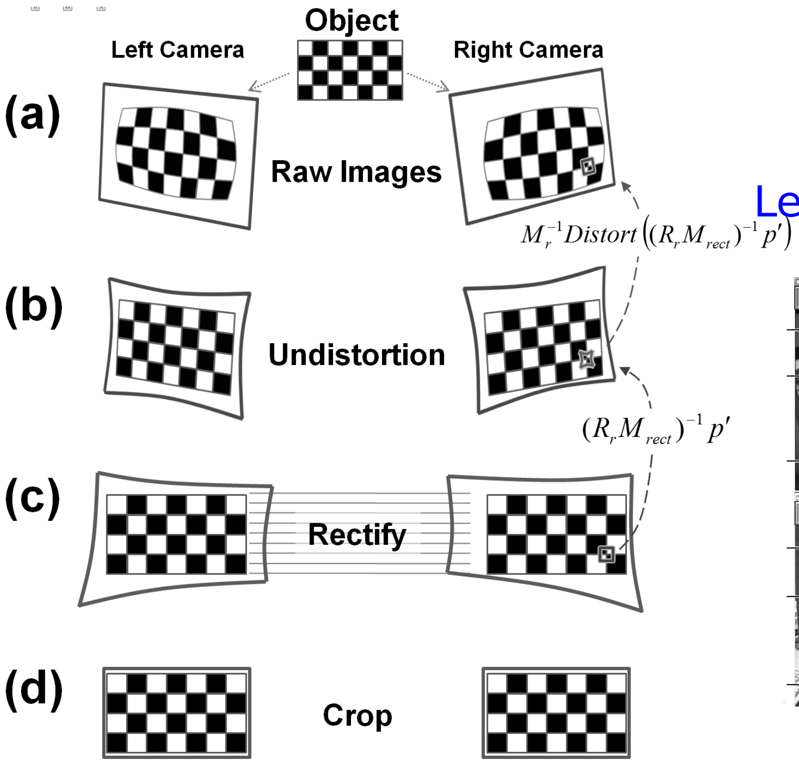


OpenCV Modules: Depth, Pose

Depth, Pose
Normals,
Planes, 3D
Features

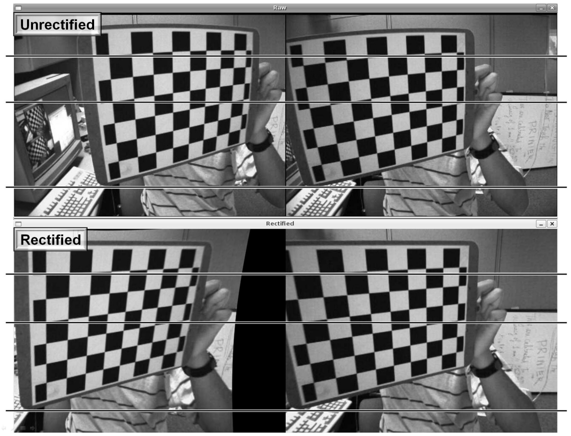


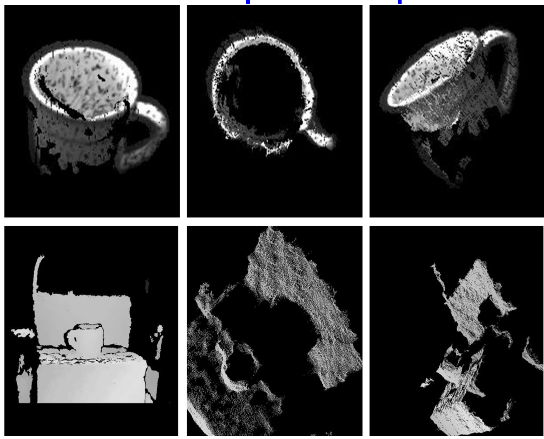


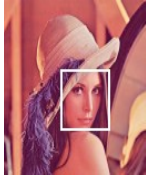


Left – right feature alignment:

Some examples of
3D stereo depth maps:







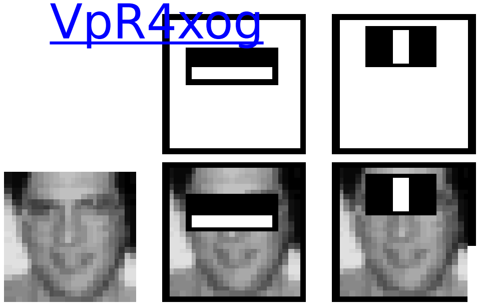
Object recognition
Machine learning

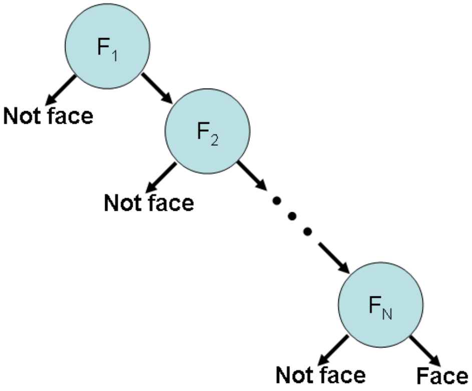
OpenCV Modules: Obj Rec/ML

https://www.youtube.com/watch?v=_RF0VpR4xog

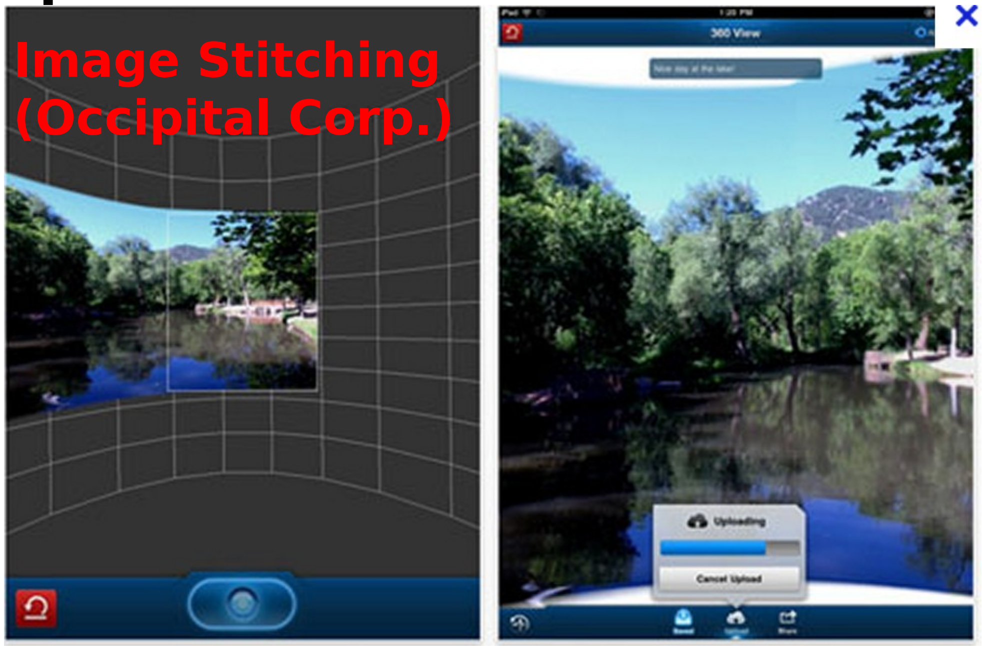
VSLAM

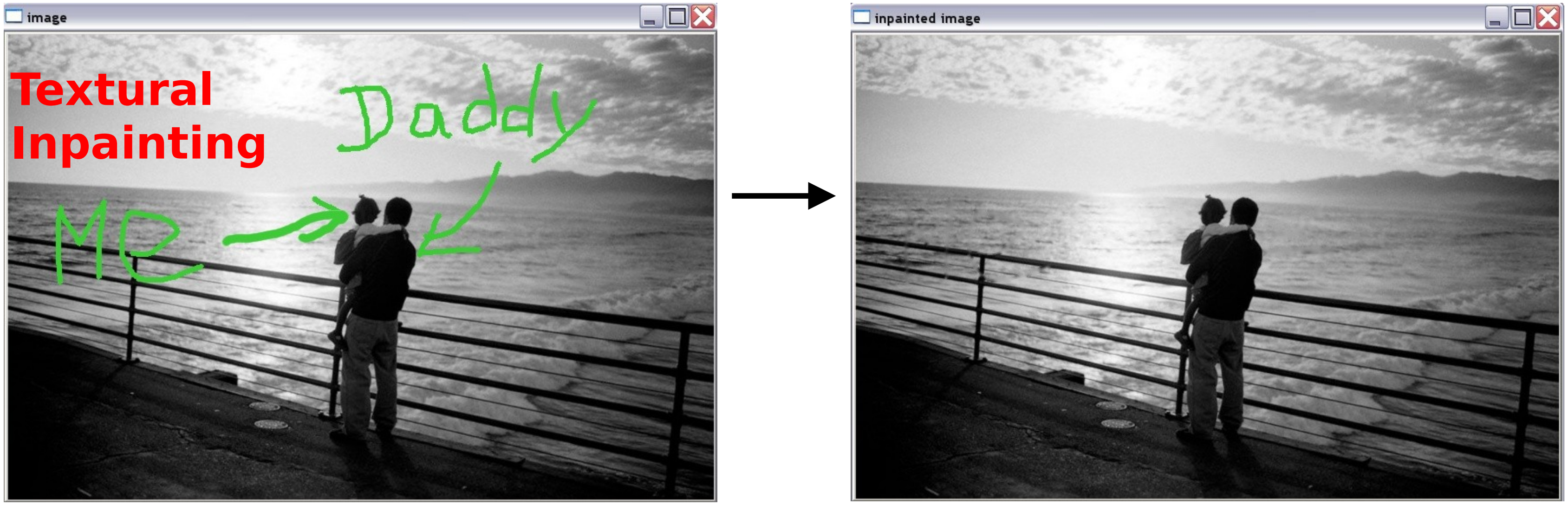
http://youtu.be/i1uUuWwbIcc





OpenCV Modules: Comp Photog





Brand New in OpenCV 3.0
User Contrib Module

- Thanks to Google Summer of Code!!
  - Supporting 15 interns!

Accepted pull requests:

xtended Python interface
D object recognition and pose
KAZE features
ar detection
omputational photography
ustom calibration and planar AR
ense optical flow

8. New line segment detector
9. Haze removal, depth estimation
10. GPU accelerated dense optical flo
11. DTAM & pose estimation
12. PNP pose detection
13. Visual saliency filters
14. Text detection and reading in wild
15. TLD tracker

OpenCV Examples

- Industrial Perception
- Magic Leap

Industrial Perception

- **Sensor driven,**
- **Real time planning**
- **Applied to distributio**

Magic Leap

- Augmented Reality done right
- Lots of computer vision (**We’re hiring**)

- Gesture recognition demo

Gesture … was going to be live demo

Questions?

Photo: Gary Bradski

Language Modules

- GPU/Cuda
- Android
- iOS
- Python
- Java

OpenCV GPU Module:

- Image processing building blocks:

Color conversions

Geometrical transforms

Per-element operations

Integrals, reductions

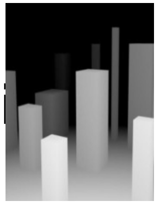
Template matching

Filtering engine

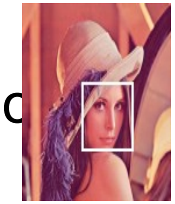
Feature detectors

- High-level algorithms:

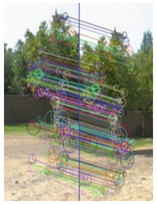
Stereo matching



Face detection



Feature matching



OpenCV GPU Module Example

```
Mat frame;
VideoCapture capture(camera);
cv::HOGDescriptor hog;

hog.setSVMDetector(cv::HOGDescriptor
::
getDefaultPeopleDetectorector());

capture >> frame;




vector<Rect> found;
hog.detectMultiScale(frame, found,
    1.4,  Size(8, 8), Size(0, 0),
1.05, 8);
```

```
Mat frame;
VideoCapture capture(camera);
cv::gpu::HOGDescriptor hog;

hog.setSVMDetector(cv::HOGDescriptor
::
getDefaultPeopleDetectorector());

capture >> frame;

GpuMat gpu_frame;
gpu_frame.upload(frame);

vector<Rect> found;
hog.detectMultiScale(gpu_frame,
found,
    1.4, Size(8, 8), Size(0, 0),
1.05, 8);
```

- Designed very similar!

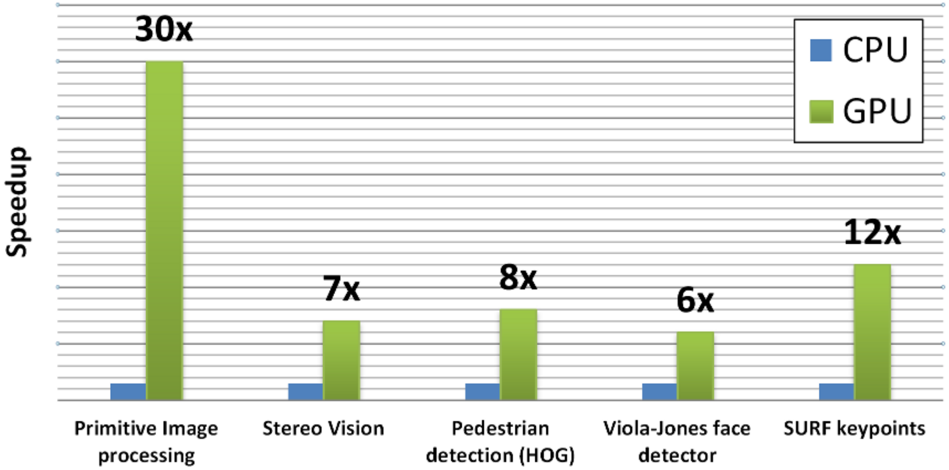
OpenCV GPU Module Performance

Tesla C2050 (Fermi) vs. Core i5-760 2.8GHz (4 cores, TBB, SSE)

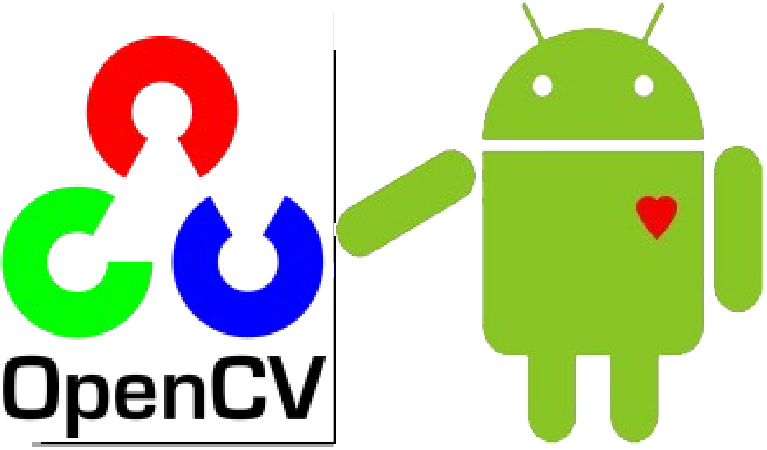
- Average speedup for primitives: 33×
  - For “good” data (large images are better)
  - Without copying to GPU

What can you get from your com

- (ance



OpenCV Android Module



- **OpenCV 2.4 for Android:**
  - **Native Android Camera Support**
  - **Multithreading**
  - **Java API (soon)**
  - **Tegra HW Optimizations (soon)**

Wiki with the latest information:

http://opencv.org/platforms/android.html

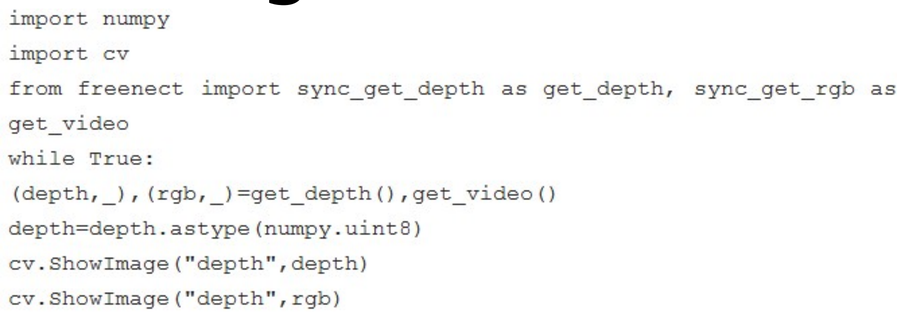
Support/discussion group:::https://groups.google.com/group/android-opencv

OpenCV iOS Module

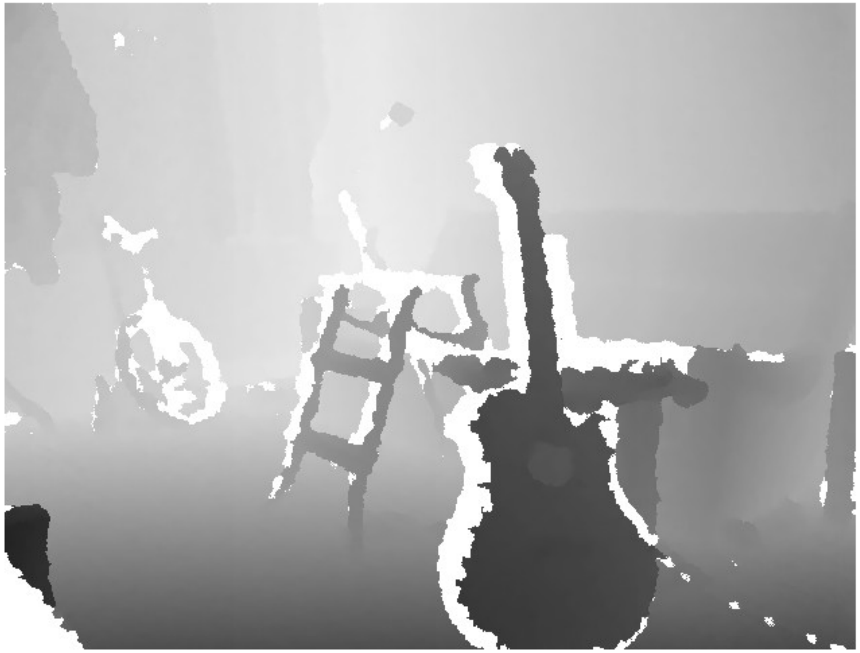
- Full support

OpenCV Python Module

- Full Python interface
- Example: Depth image from Kinect:

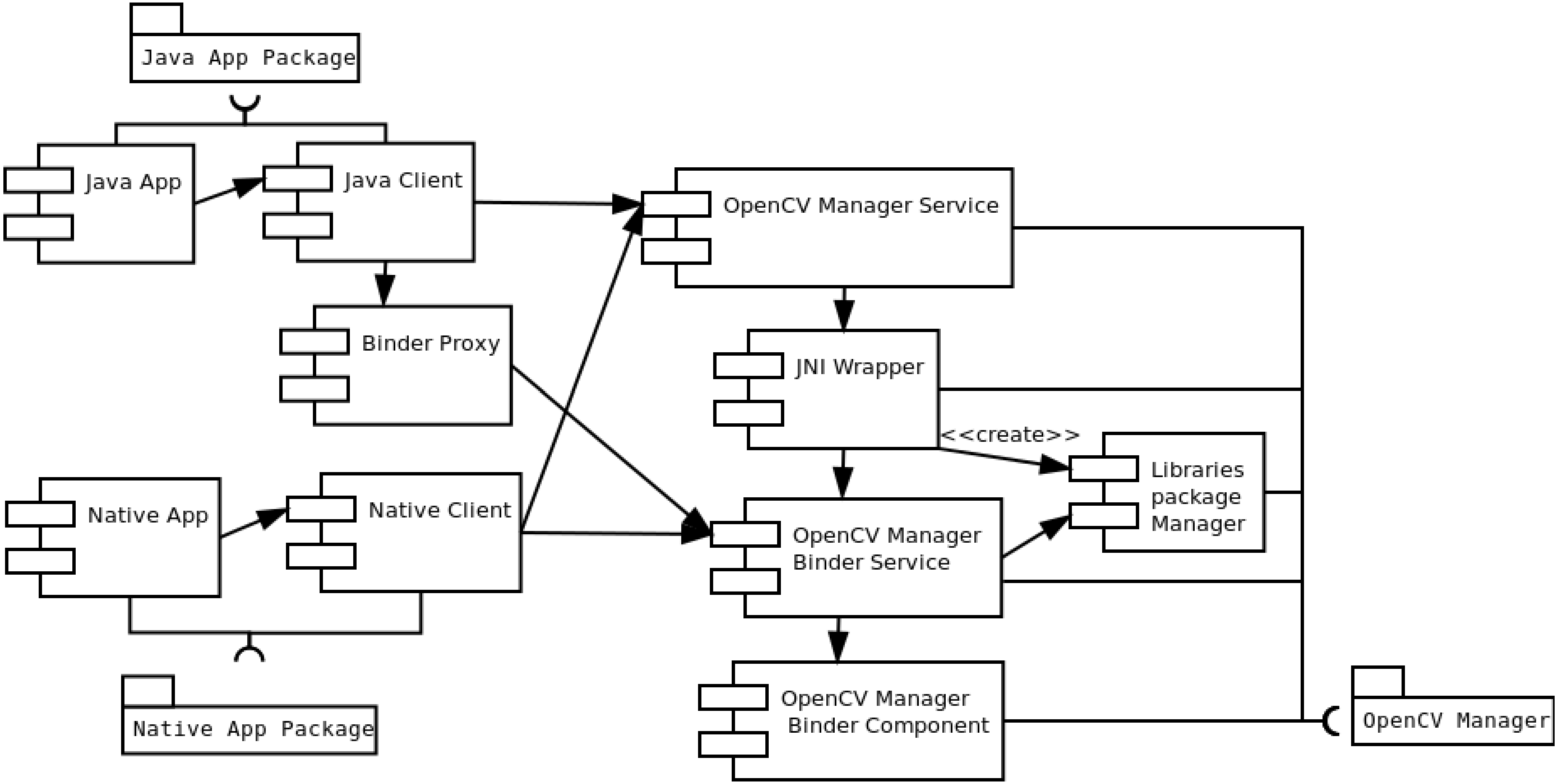


```
import numpy
import cv
from freenect import sync_get_depth as get_depth, sync_get_rgb as
get_video
while True:
(depth,_),(rgb,_)=get_depth(),get_video()
depth=depth.astype(numpy.uint8)
cv.ShowImage("depth",depth)
cv.ShowImage("depth",rgb)
```



Depth image

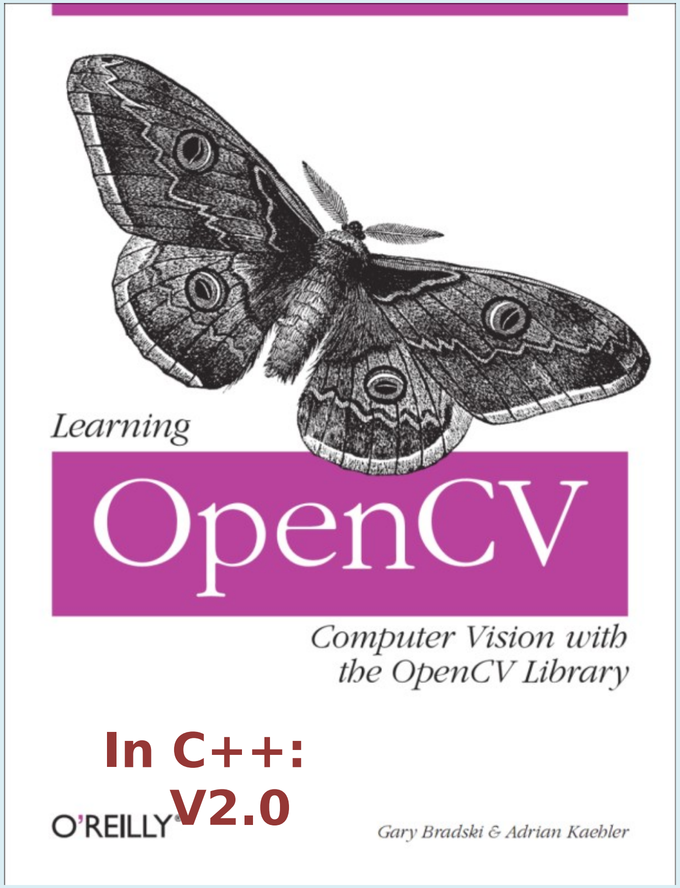
OpenCV Java Module



Book and Foundation

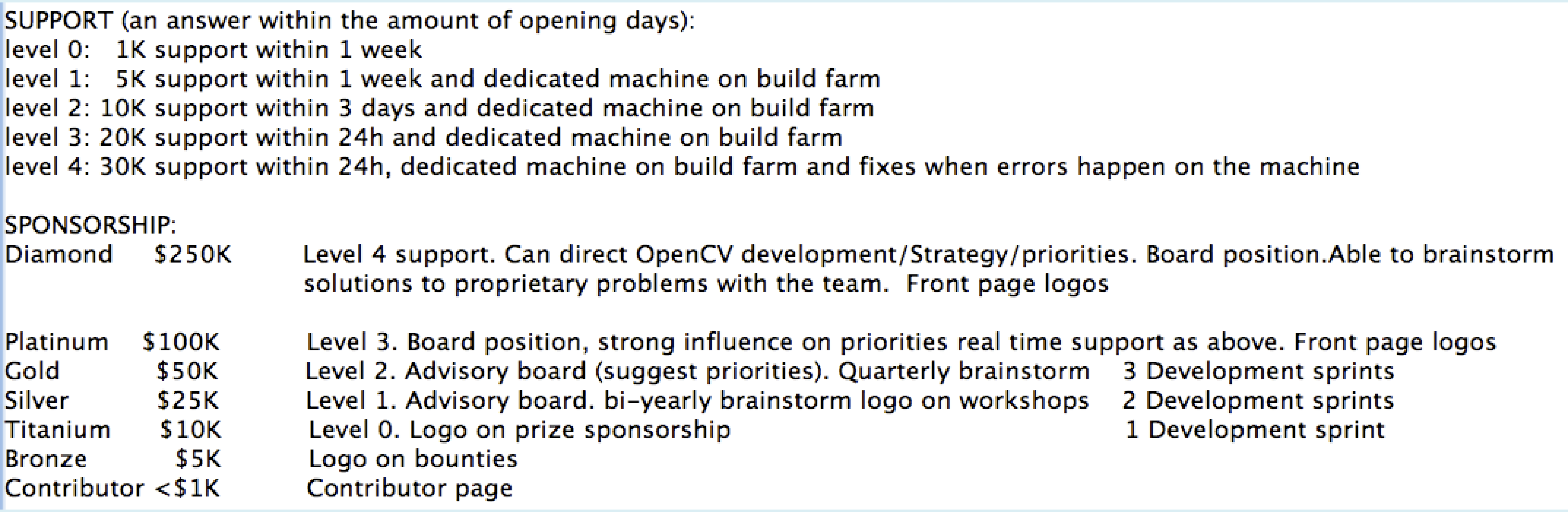
Learning OpenCV V2.0

- Out in Summer 2014!


Learning
OpenCV
Computer Vision with the OpenCV Library
O'REILLY
Gary Bradski & Adrian Kaehler

In C++: V2.0

OpenCV Foundation Support



SUPPORT (an answer within the amount of opening days):
level 0: 1K support within 1 week
level 1: 5K support within 1 week and dedicated machine on build farm
level 2: 10K support within 3 days and dedicated machine on build farm
level 3: 20K support within 24h and dedicated machine on build farm
level 4: 30K support within 24h, dedicated machine on build farm and fixes when errors happen on the machine

SPONSORSHIP:

Diamond	$250K	Level 4 support. Can direct OpenCV development/Strategy/priorities. Board position.Able to brainstorm solutions to proprietary problems with the team. Front page logos	
Platinum	$100K	Level 3. Board position, strong influence on priorities real time support as above. Front page logos	
Gold	$50K	Level 2. Advisory board (suggest priorities). Quarterly brainstorm	3 Development sprints
Silver	$25K	Level 1. Advisory board. bi-yearly brainstorm logo on workshops	2 Development sprints
Titanium	$10K	Level 0. Logo on prize sponsorship	1 Development sprint
Bronze	$5K	Logo on bounties	
Contributor	<$1K	Contributor page	