Intro to the OpenCV Library

for Computer Vision lectures and
Introduction to CV and other CV lab classes

(some slides are cc from 'opencv 3.0' Kirill Kornyakov, Itseez)

Topics

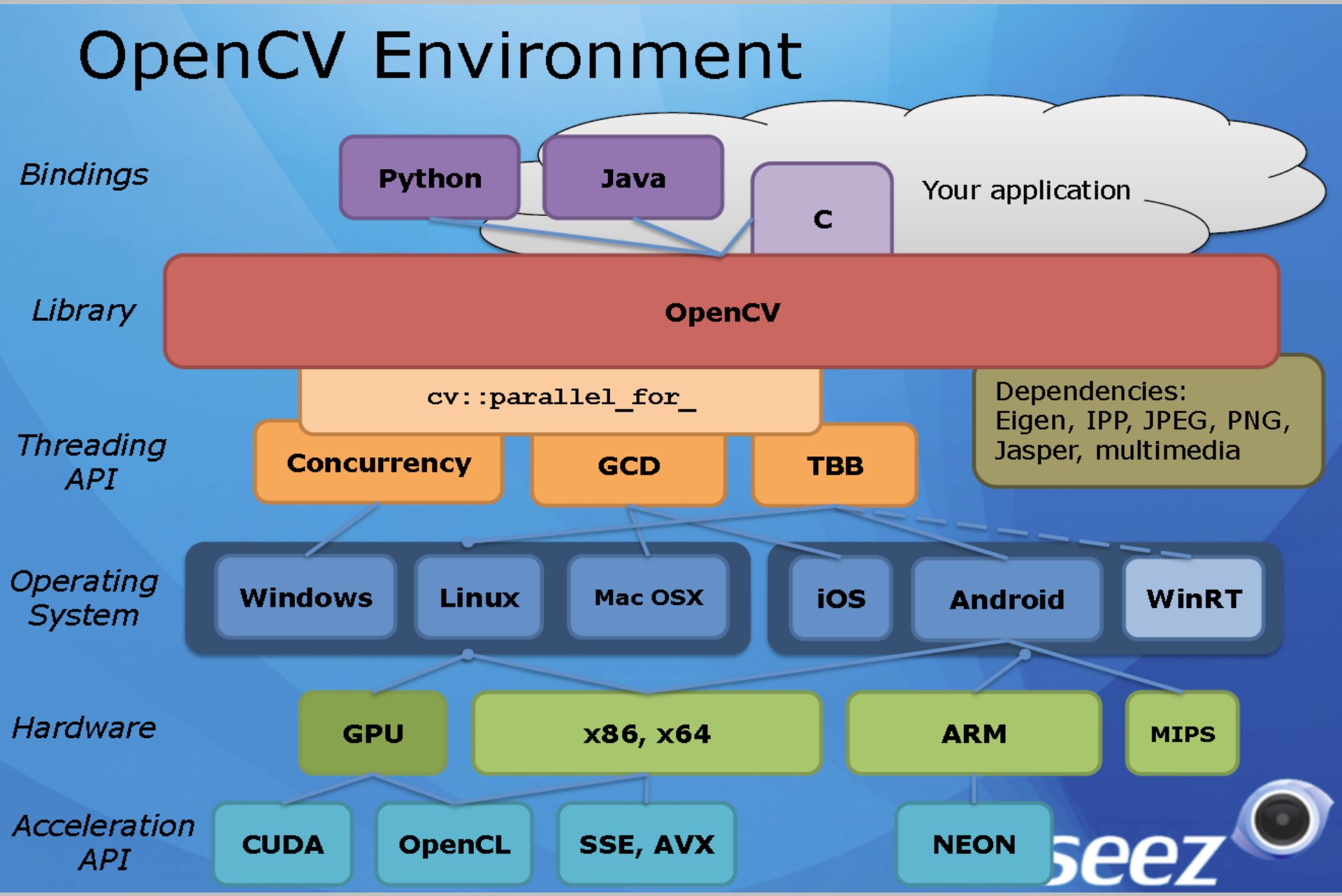
1. Why
2. What
3. Install
4. Example Project
5. Your Task
6. Your Questions

Why OpenCV?

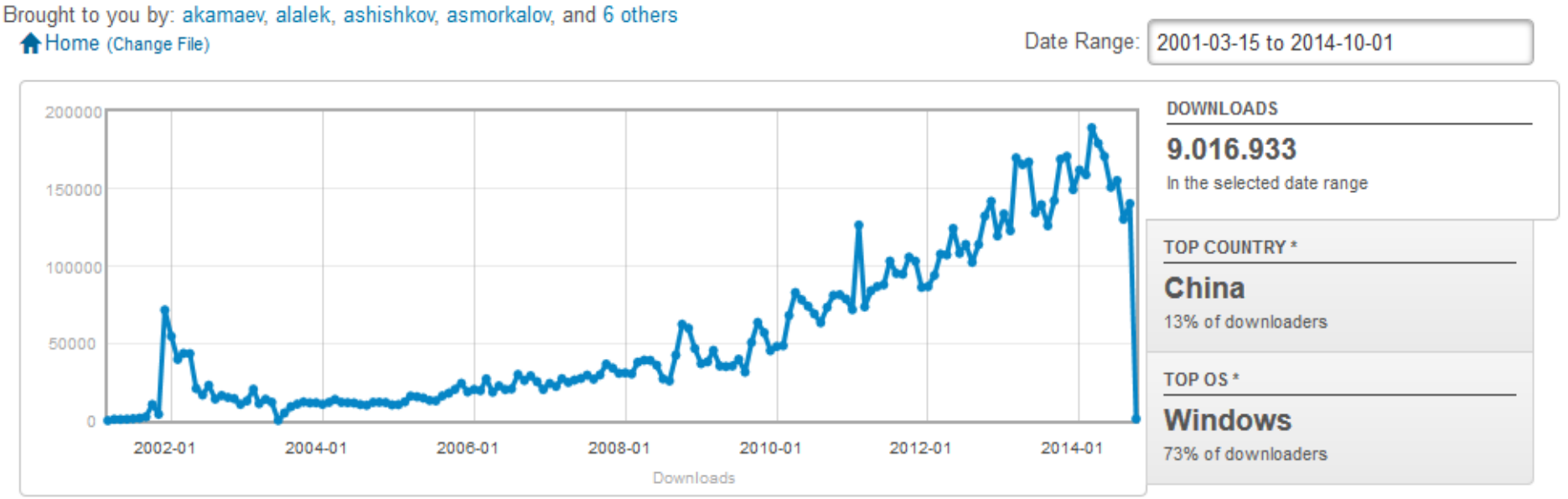
1. 2,500+ algorithms and functions
2. Cross-platform, portable API
3. Real-time performance
4. Liberal BSD license
5. fast and regular updates




OpenCV Environment

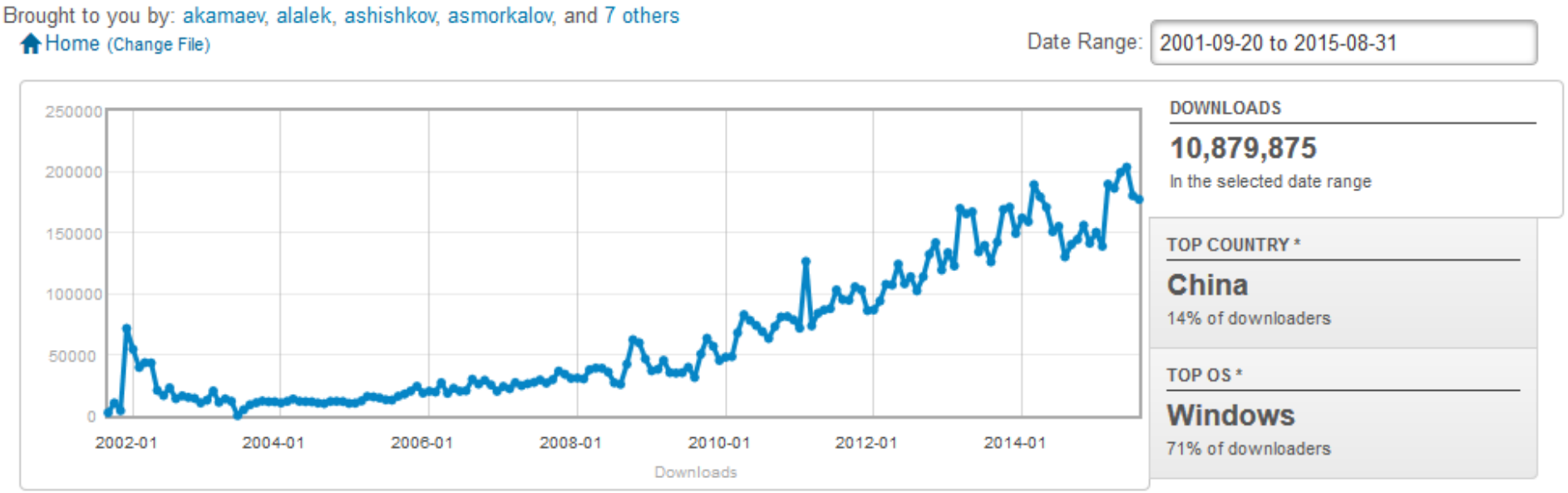


History



Intel → 2001 open src, 2008: Willow Garage, Itseez, 2010: Nvidia

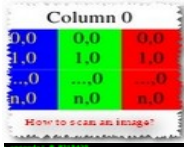
History



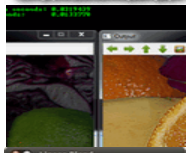
Intel → 2001 open src, 2008: Willow Garage, Itseez, 2010: Nvidia

What? core module tutorials:

Mat - The Basic Image Container



How to scan images, lookup tables and time measurement with OpenCV



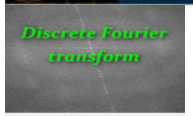
Mask operations on matrices

Adding (blending) two images using OpenCV

Changing the contrast and brightness of an image!

Basic Drawing

Random generator and text with OpenCV



Discrete Fourier Transform

File Input and Output using XML and YAML files

What? imgproc module tutorials:

Smoothing Images
Eroding and Dilating
More Morphology Transformations
Image Pyramids
Basic Thresholding Operations
Making your own linear filters!
Adding borders to your images
Sobel Derivatives
Laplace Operator
Canny Edge Detector
Hough Line Transform
Hough Circle Transform
Remapping
Affine Transformations

Histogram Equalization
Histogram Calculation
Histogram Comparison
Back Projection
Template Matching
Finding contours in your image
Convex Hull
Creating Bounding boxes and circles for contours
Creating Bounding rotated boxes and ellipses for contours
Image Moments
Point Polygon Test

What? Other modules:

Highgui:
Adding a Trackbar to our applications!
Video Input with OpenCV and similarity measurement
Creating a video with OpenCV

calib3d:
Camera calibration

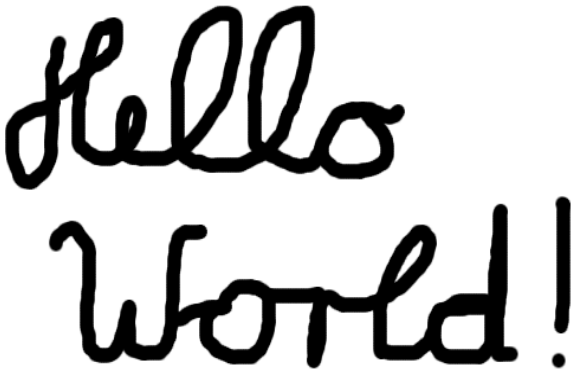
ml:
Introduction to Support Vector Machines
Support Vector Machines for non-lin. Separable Data

feature2d:
Harris corner detector
Shi-Tomasi corner detector
Creating your own corner detector
Detecting corners location in subpixeles
Feature Description
Feature Matching with FLANN
Features2D + Homography to find a known object
Detection of planar objects

objdetect:
Cascade Classifier

What? code examples:

```
int main(int argc, char** argv)
{
    Mat img = imread(argv[1], 1);
    imshow("", img);
    waitKey();
    return 0;
}
```



What? code examples:

```
int main(int argc, char** argv)
{
    Mat img, gray;
    img = imread(argv[1], 1);
    imshow("original", img);
    cvtColor(img, gray, COLOR_BGR2GRAY);
    GaussianBlur(gray, gray, Size(7, 7),
                 1.5);
    Canny(gray, gray, 0, 50);
    imshow("edges", gray);
    waitKey();
    return 0;
}
```

What? code examples:

Threshold:

```
Mat emptyPixImg = GrayImg < 1;
```

Image from (Camera- or) Directory-stream:

```
VideoCapture cap("TextureImages/Texture_%02d_inpaint.png");
Mat Img;
cap >> Img;
```

Create a 2D-Gaussian:

```
Mat Gauss2D = Mat::zeros(TemplateWidth, TemplateWidth,
                                                    CV_32FC1);
Gauss2D.at<float>( TemplateHW, TemplateHW) = 1.0;
GaussianBlur(Gauss2D, Gauss2D, Size(TemplateWidth,
                                TemplateWidth), sigma, sigma);
```

What? code examples:

pointer work to speed up inner loops:

(1)

```
int** iim = new int*[h];
for (y=0; y<h; y++)
{  iim[y] = IntegralImg.ptr<int>(y);
}
int diffy = 2*( iim[y][x+dx]    - iim[y][x-dx] )  +
                iim[y-dy][x-dx] - iim[y-dy][x+dx] +
                iim[y+dy][x-dx] – iim[y+dy][x+dx];
```

(2)

```
float *pCR, *pCRData = (float*) CorrResult.data;
*pCR  = pCRData + y*w;
for ( int x = TemplateWidth; x < w-TemplateWidth; x++ )
{  pCR[x] = ssd;    // write ssd result to result image
}
```

How?

1. Home: opencv.org
2. Documentation: docs.opencv.org
3. Q&A forum: answers.opencv.org
4. Report issues: code.opencv.org
5. Develop: https://github.com/Itseez/opencv

How? Install (linux):

1. download:https://github.com/Itseez/opencv/archive/3.0.0.zip
2. run Cmake(gui), check/install add-ons and configure until all problems have gone generate
3. make
4. sudo make install

5. setup your ide
6. run example

How? Install (linux):

1. download:https://github.com/Itseez/
2. run Cmake(gui), check/install add-ons and configure until all problems have gone generate
3. make
4. sudo make install

now:

5. setup your ide
6. run example

Our plans

1. Set up development environment and
   make a simple segmentation program.
2. You are free to use opencv and other example code you find,
3. but have to
   - put it all together on your own
   - cite your source in a comment.
4. Good C++ coding style and a lot of comments!
5. Send your results the day before the next task starts
   to holger.heidrich, first task within 13 days.
6. Send source code and header files (no project files) as well
   as input (if not given) and result images.
7. Your code must compile without errors on Win
   and Linux systems (i.e. avoid Win-specific code).

Your first task: superpixel & colour reduction

1. Install OpenCV with debug libs on your system.
2. In a given colour Image, find “superpixel” regions:
   - calculate the magnitude of the colour gradient (x+y direction)
   - put them (together with pixel number) in a priority queue
   - (1) pop lowest gradient pixel
   - 4-connect the current pixel to all neighbours with colour distance lower a given Δ (changeable by slider)
   - remove all connected pixels from the priority queue
   - if priority queue not empty go to (1)
   - colour each region with its mean colour
   - output the resulting image
   - discuss problems and possible refinements of this algorithm

Hint:
Search opencv\samples\cpp*.cpp containing keywords you need (gradient, region, mask …)
see also core module tutorials

Credits:

- no compilation errors,
- solves the task,
- hand in in time

→ 1 Point out of min. 8 you need to pass the CV1 exercise course.

The exercise is part of the exam; if you got at least 8 points
the questions that regard the exercise will concentrate around what you did,
otherwise they will cover the whole set of exercise tasks.)