OpenCV

Open Source **C**omputer **V**ision

Anja Groch
Medical and Biological Informatics, DKFZ Heidelberg, Germany

dkfz.
DEUTSCHES
KREBSFORSCHUNGSZENTRUM
IN DER HELMHOLTZ-GEMEINSCHAFT

Overview

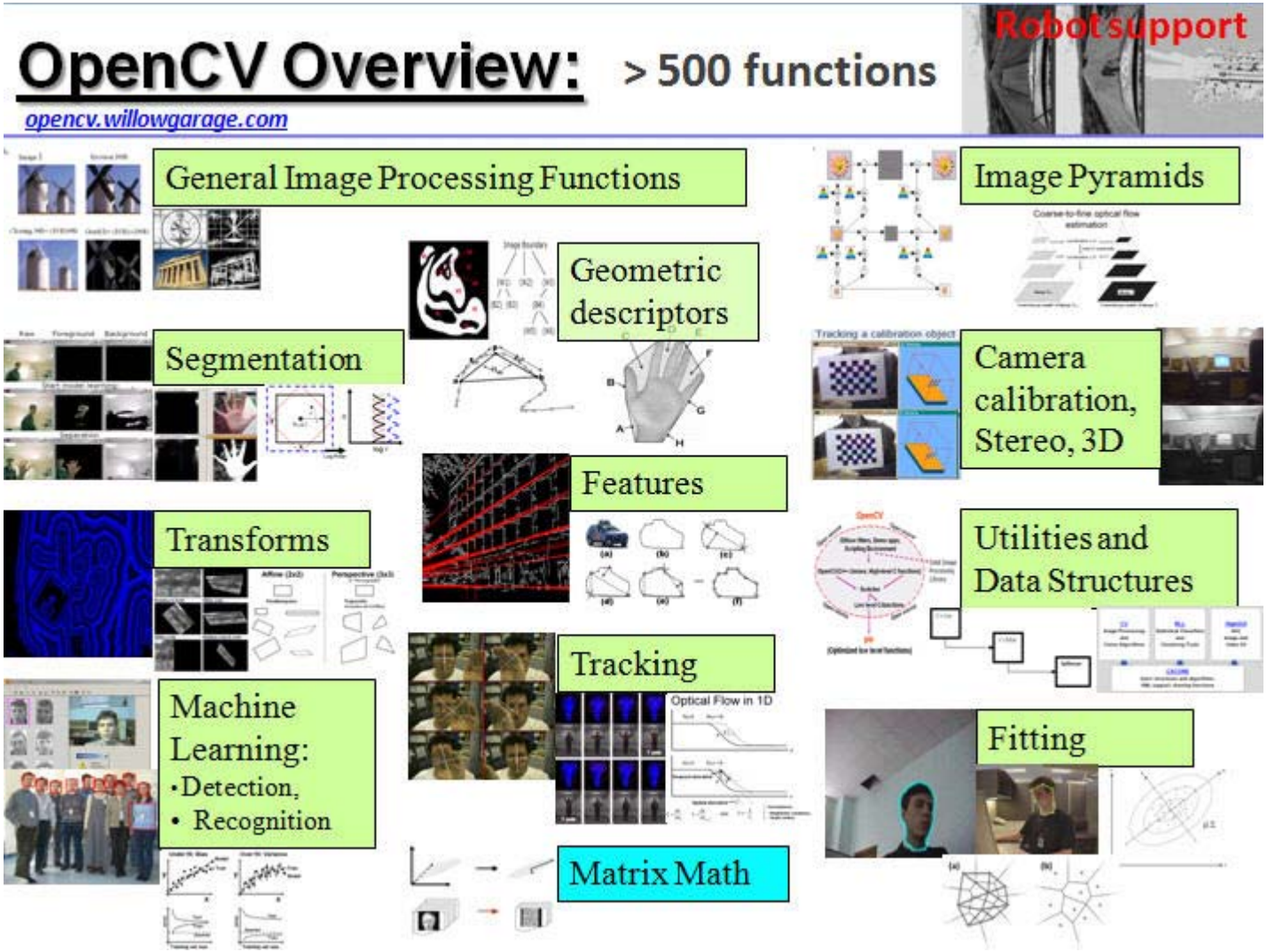
- Library for **real time computer vision**
- Founded at Intel, now supported from Willow Garage, company for robotic applications
- Open Source

- Under Windows, Unix and Mac OS X
- Written in C and C++
- Development of interfaces for Python, Ruby, Matlab, …

- Can take advantage of multicore processors
- GPU programming

- Current release version 2.2
  - ➢ Huge improvements from 1.x to 2.x
    - – in documentation
    - – new C++-interface and thereby reducing of programming errors

Overview


OpenCV Overview:
> 500 functions
opencv.willowgarage.com
Robot support
General Image Processing Functions
Image Pyramids
Geometric descriptors
Segmentation
Camera calibration, Stereo, 3D
Features
Transforms
Utilities and Data Structures
Tracking
Optical Flow in 1D
Machine Learning:
•Detection,
• Recognition
Fitting
Matrix Math

OpenCV vs ITK

Different focus

ITK: Basic image processing, segmentation, registration for medical image data (2D, 3D, …)

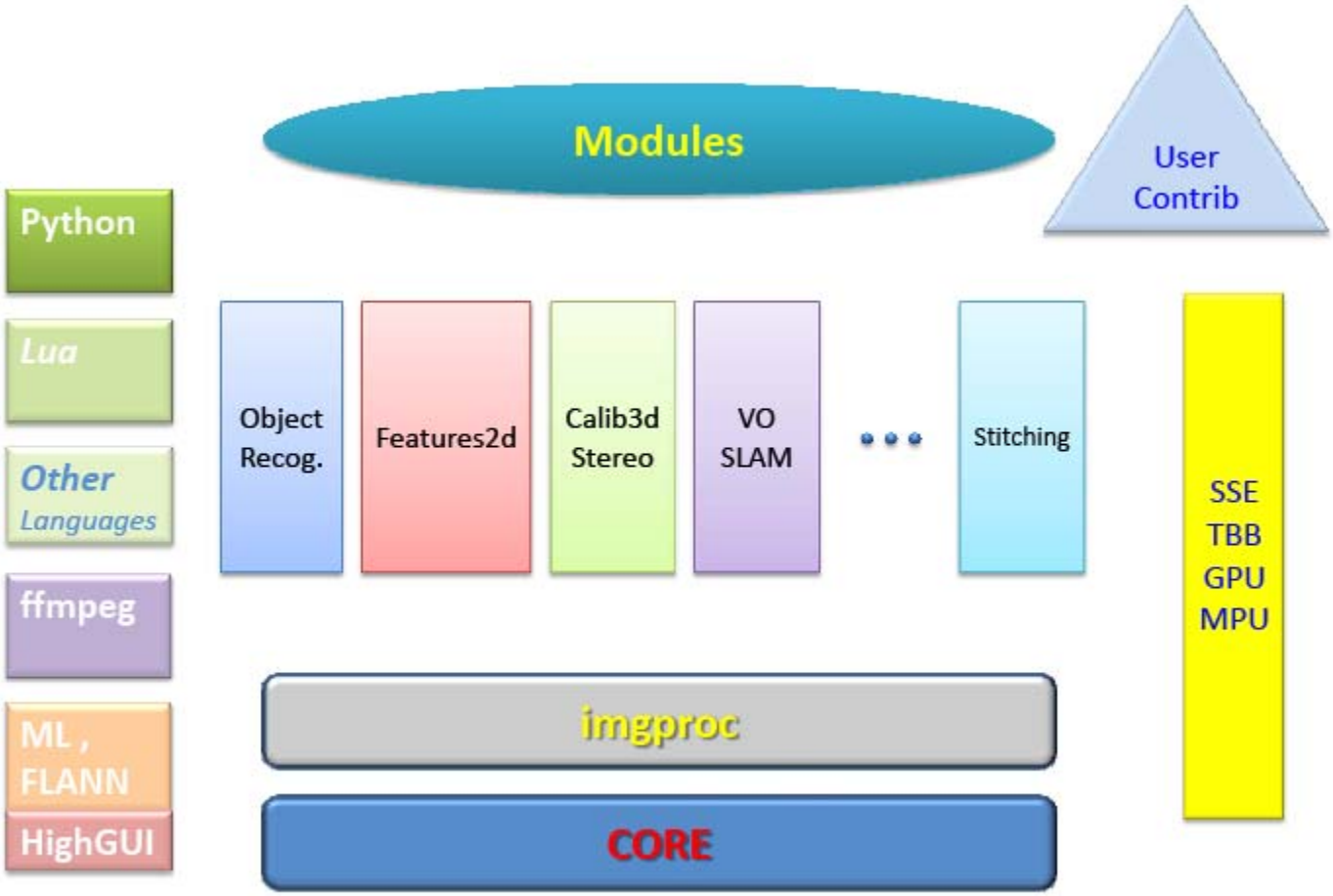
OpenCV: Computer vision in general, basic image processing part for 2D images PLUS machine learning

In case of algorithms, provided by both, OpenCV and ITK

ITK: Better integration in MITK
Filter structure, smart pointer, …

OpenCV: Faster

OpenCV Conceptual Stucture



Main structures

Key OpenCV Classes

`Point_`	Template 2D point class
`Point3_`	Template 3D point class
`Size_`	Template size (width, height) class
`Vec`	Template short vector class
`Scalar`	4-element vector
`Rect`	Rectangle
`Range`	Integer value range
`Mat`	2D dense array (used as both a matrix or an image)
`MatND`	Multi-dimensional dense array
`SparseMat`	Multi-dimensional sparse array
`Ptr`	Template smart pointer class

Simple Image Processing

`filter2D()`	Non-separable linear filter
`sepFilter2D()`	Separable linear filter
`boxFilter()`, `GaussianBlur()`, `medianBlur()`, `bilateralFilter()`	Smooth the image with one of the linear or non-linear filters
`Sobel()`, `Scharr()`	Compute the spatial image derivatives
`Laplacian()`	compute Laplacian: $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$
`erode()`, `dilate()`	Erode or dilate the image

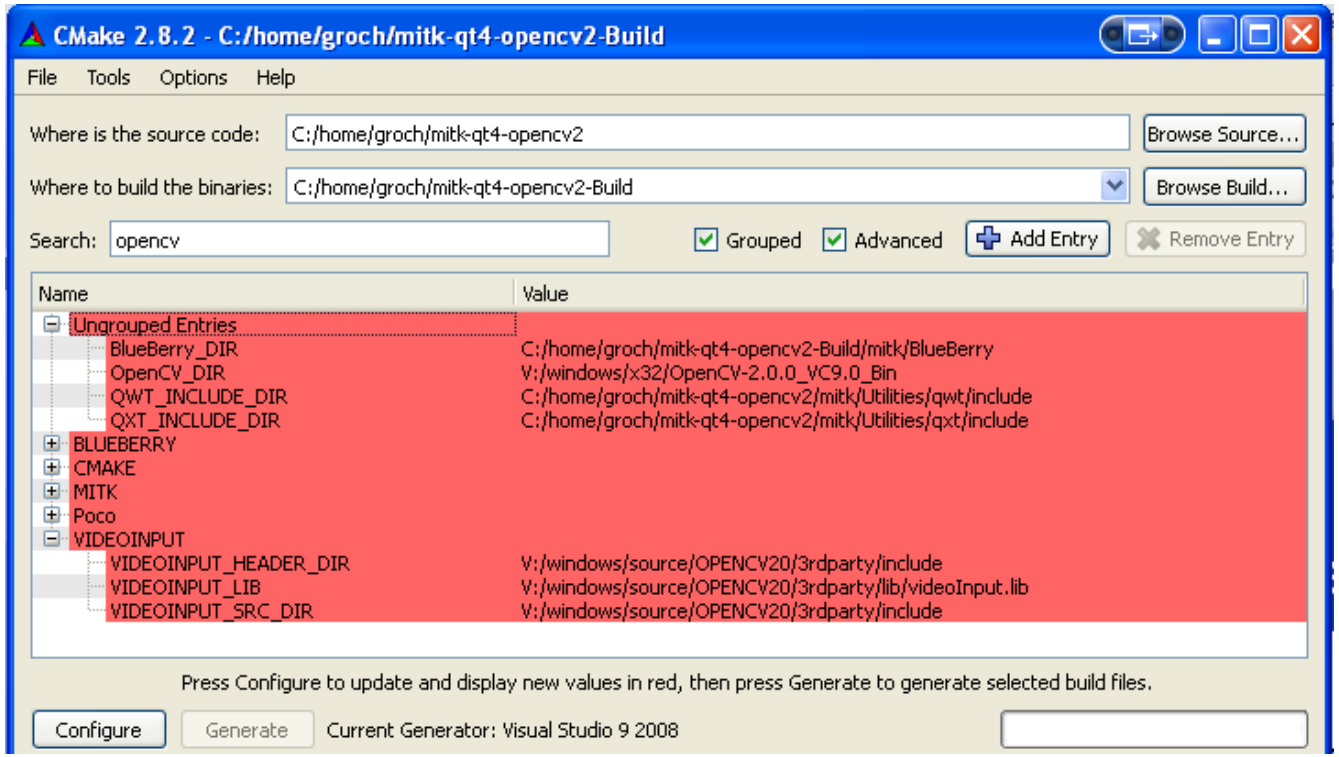
Camera Calibration, Pose, Stereo

`calibrateCamera()`	Calibrate camera from several views of a calibration pattern.
`findChessboardCorners()`	Find feature points on the checkerboard calibration pattern.
`solvePnP()`	Find the object pose from the known projections of its feature points.
`stereoCalibrate()`	Calibrate stereo camera.
`stereoRectify()`	Compute the rectification transforms for a calibrated stereo camera.
`initUndistortRectifyMap()`	Compute rectification map (for `remap()`) for each stereo camera head.
`StereoBM`, `StereoSGBM`	The stereo correspondence engines to be run on rectified stereo pairs.
`reprojectImageTo3D()`	Convert disparity map to 3D point cloud.
`findHomography()`	Find best-fit perspective transformation between two 2D point sets.

To calibrate a camera, you can use `calibration.cpp` or `stereo_calib.cpp` samples. To get the disparity maps and the point clouds, use `stereo_match.cpp` sample.

Integration into MITK

- Download source data from here: http://sourceforge.net/projects/opencvlibrary
- Build binaries for your system
- Integrate into MITK:



And a lot more...

OpenCV Wiki:
http://opencv.willowgarage.com/wiki

OpenCV Code Repository:
http://sourceforge.net/projects/opencvlibrary/

New Book on OpenCV:
http://oreilly.com/catalog/9780596516130/

Code examples from the book:
http://examples.oreilly.com/9780596516130/

User Group:
http://tech.groups.yahoo.com/group/OpenCV/join