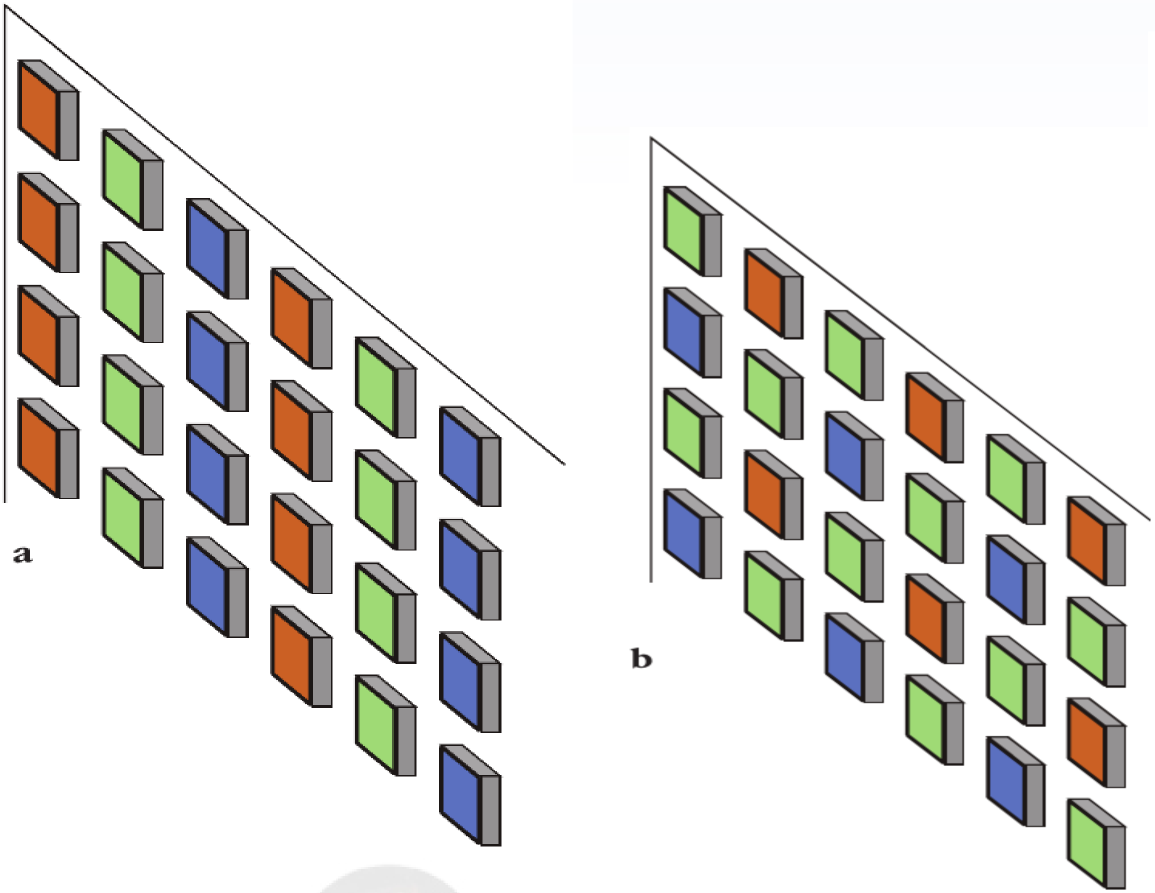
BSC~Microsoft Research Centre

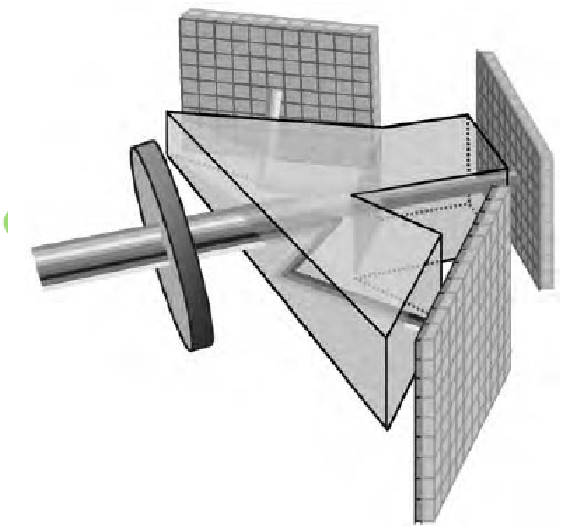
Tassadaq Hussain

Microsoft Barcelona Supercomputing Center
Universitat Politécnica de Catalunya
Barcelona, Spain

UCERD
Gathering Intellectuals
www.ucerd.com

Three-chip color Camera

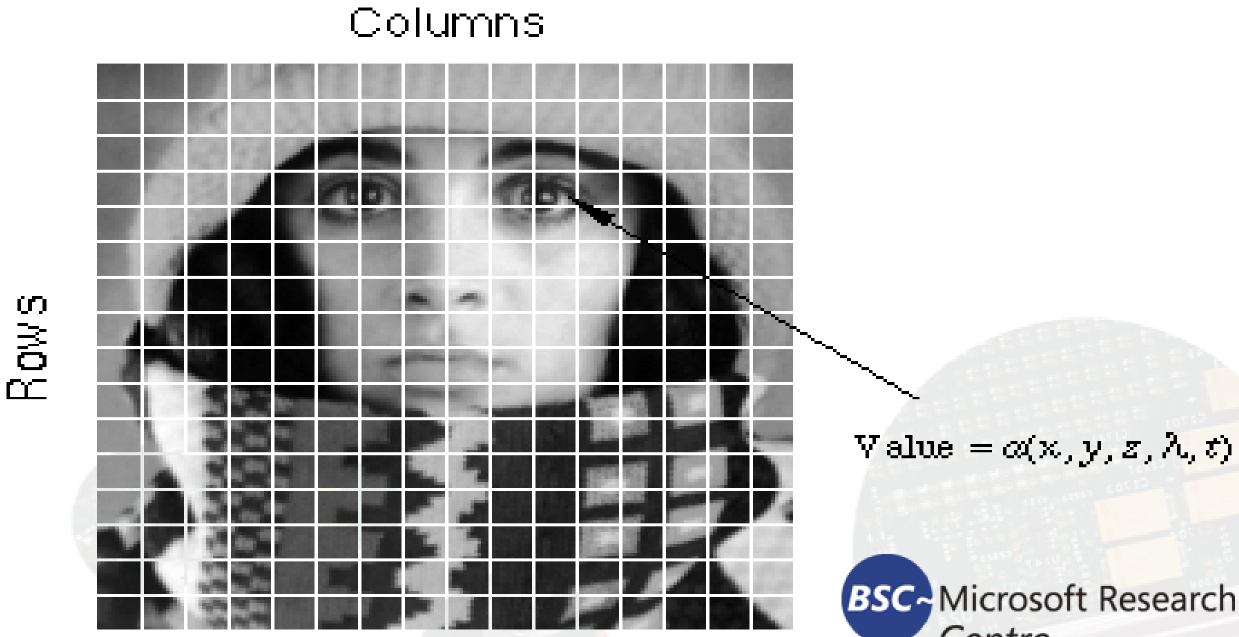




(a) Bayer (b) Filter patterns used in single chip cameras.

Color Pixel = **Red** (8bit) + **Green** (8bit) + **Blue** (8bit)

Gray scale intensity = 0.299 R + 0.587 G + 0.114 B



Pixel >> Image >> Video

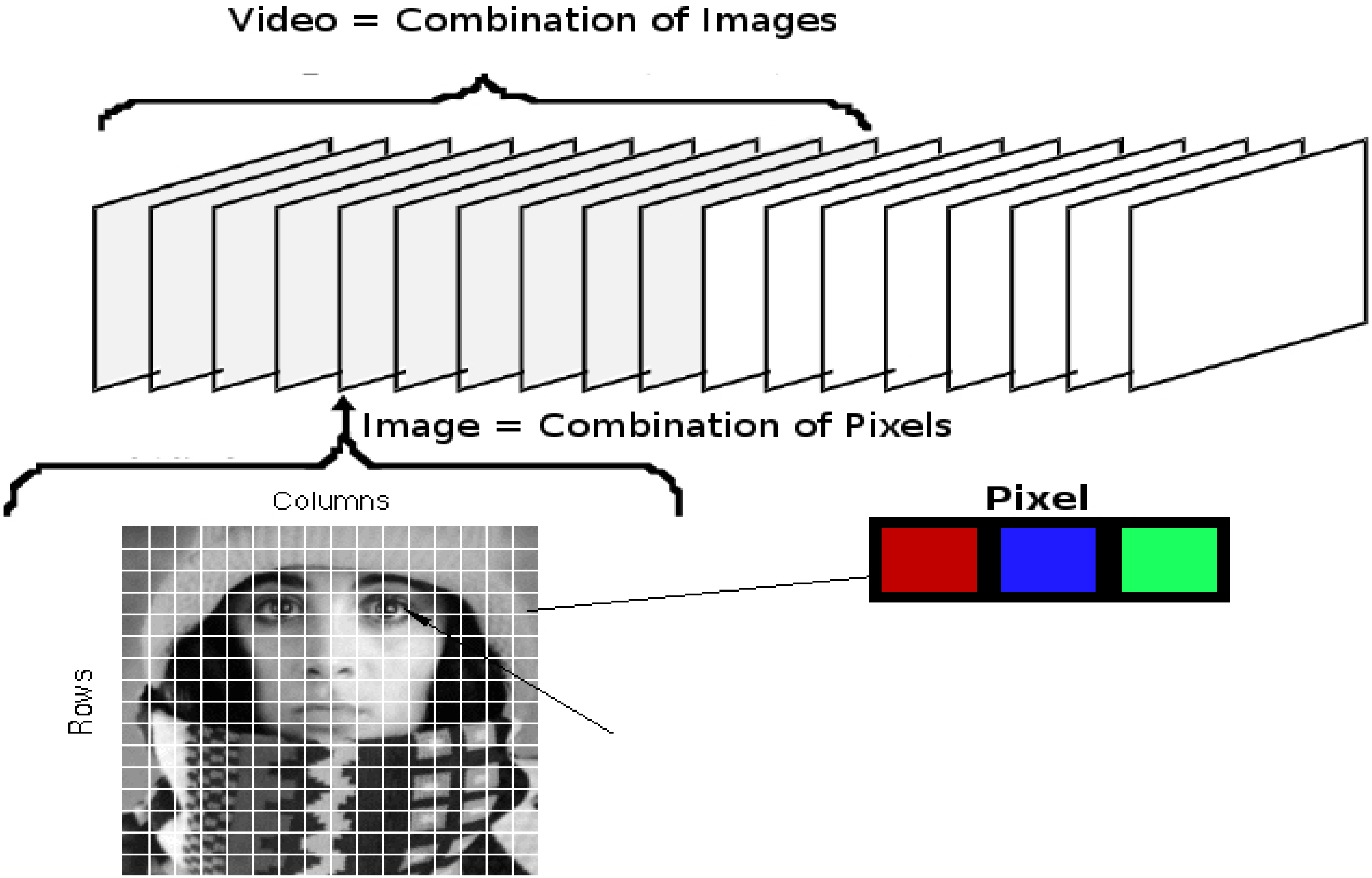
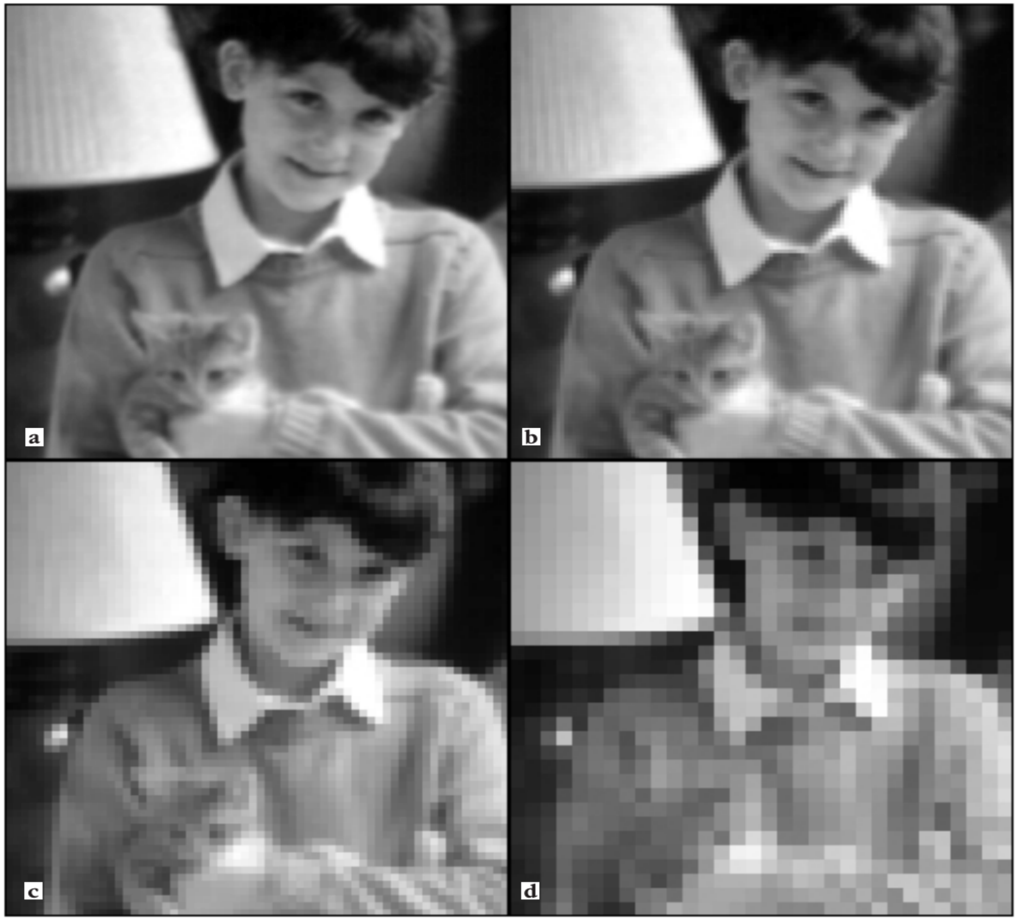


Image Resolution



(a) 256 × 256; (b) 128 × 128; (c) 64 × 64, (d)32 × 32.

Pixel Depth

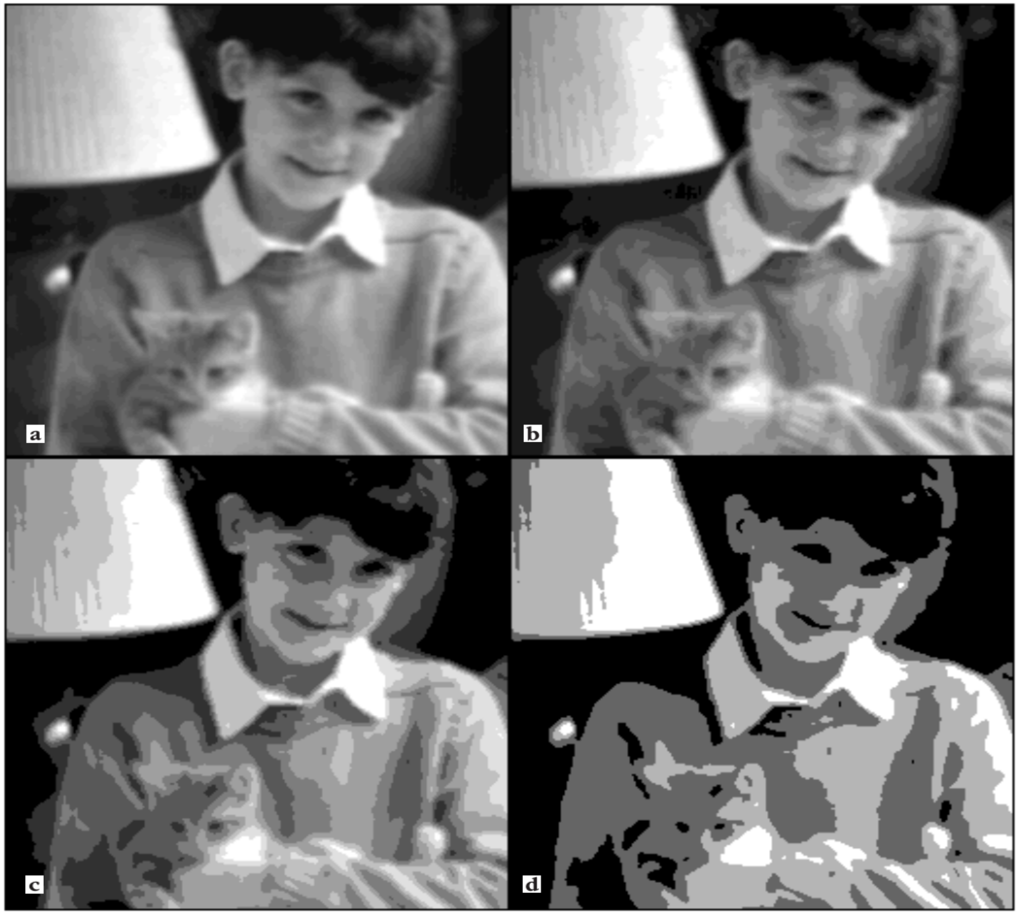


Image 256x256 array pixels: (a) 32 bit (b) 16 (c) 8 (d) 4

Performance Measures

- 3 Mega Pixel Image = 3145720 pixels
- A 32 bit Processor = 3.14 million operation / sec

  Pixels = 2048 x 1536 x 24 bits/pixel
- Local Memory = 9.4 Mega Byte for single Image
- Video Processing = $3.14 \times 10^6 \times 30$ (fps)

  $= 94.2 \times 10^6$

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

Processor / System	Dhrystone MIPS / MIPS
Nios II	190 MIPS at 165 MHz
ARM Cortex A7	2,850 MIPS at 1.5 GHz
ARM Cortex-A9 (Dual core)	7,500 MIPS at 1.5 GHz
Raspberry Pi 2	1186 MIPS per core at 1.0 GHz
Nvidia Tegra 3 (Quad core Cortex-A9)	13,800 MIPS at 1.5 GHz
Intel Core 2 Extreme QX6700 (Quad core)	49,161 MIPS at 2.66 GHz
Intel Core i7 920 (Quad core)	82,300 MIPS at 2.93 GHz

Simple Thresholding

```
Read Image Pixel // I/O Operation
if(pix_value>value) // Branch Operation
pix_value=value // Assignment Operation
```

A 3 Mega Pix Image requires

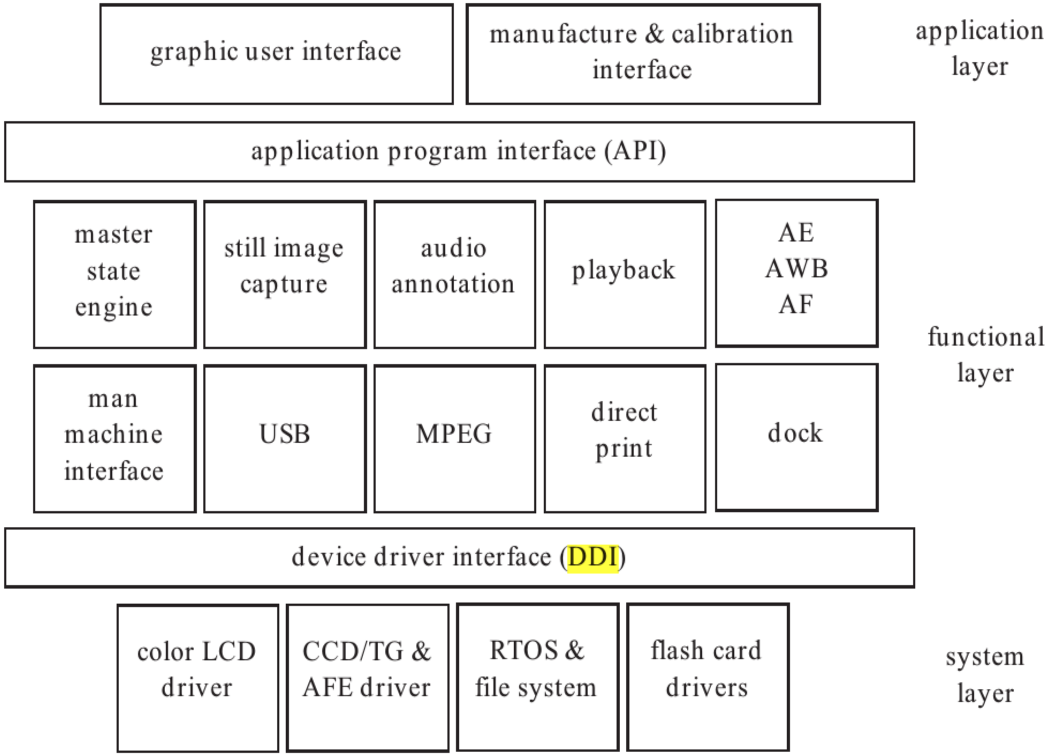
2048 x 1536 Input/Output Operations

2048 x 1536 Branch Operations

2048 x 1536 Assignment Operations

Total = 2048 x 1536 x 3 = 9 Million Operations

Software Platform of Digital Camera



graphic user interface

manufacture & calibration interface

application layer

application program interface (API)

master state engine

still image capture

audio annotation

playback

AE AWB AF

man machine interface

USB

MPEG

direct print

dock

functional layer

device driver interface (DDI)

color LCD driver

CCD/TG & AFE driver

RTOS & file system

flash card drivers

system layer

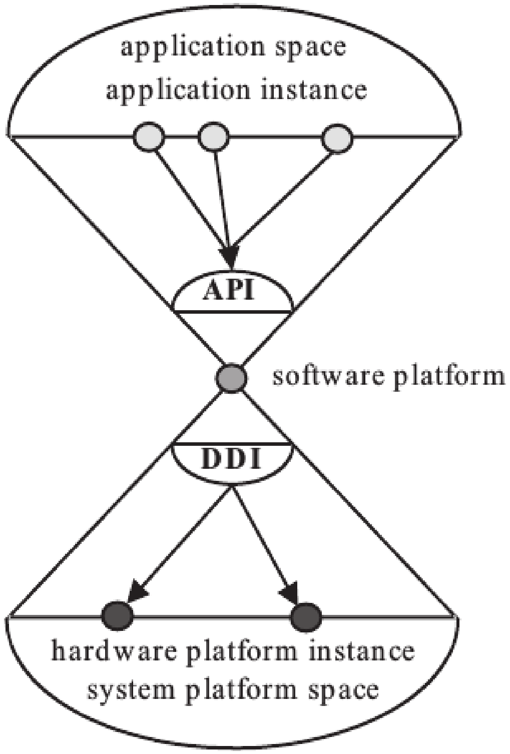
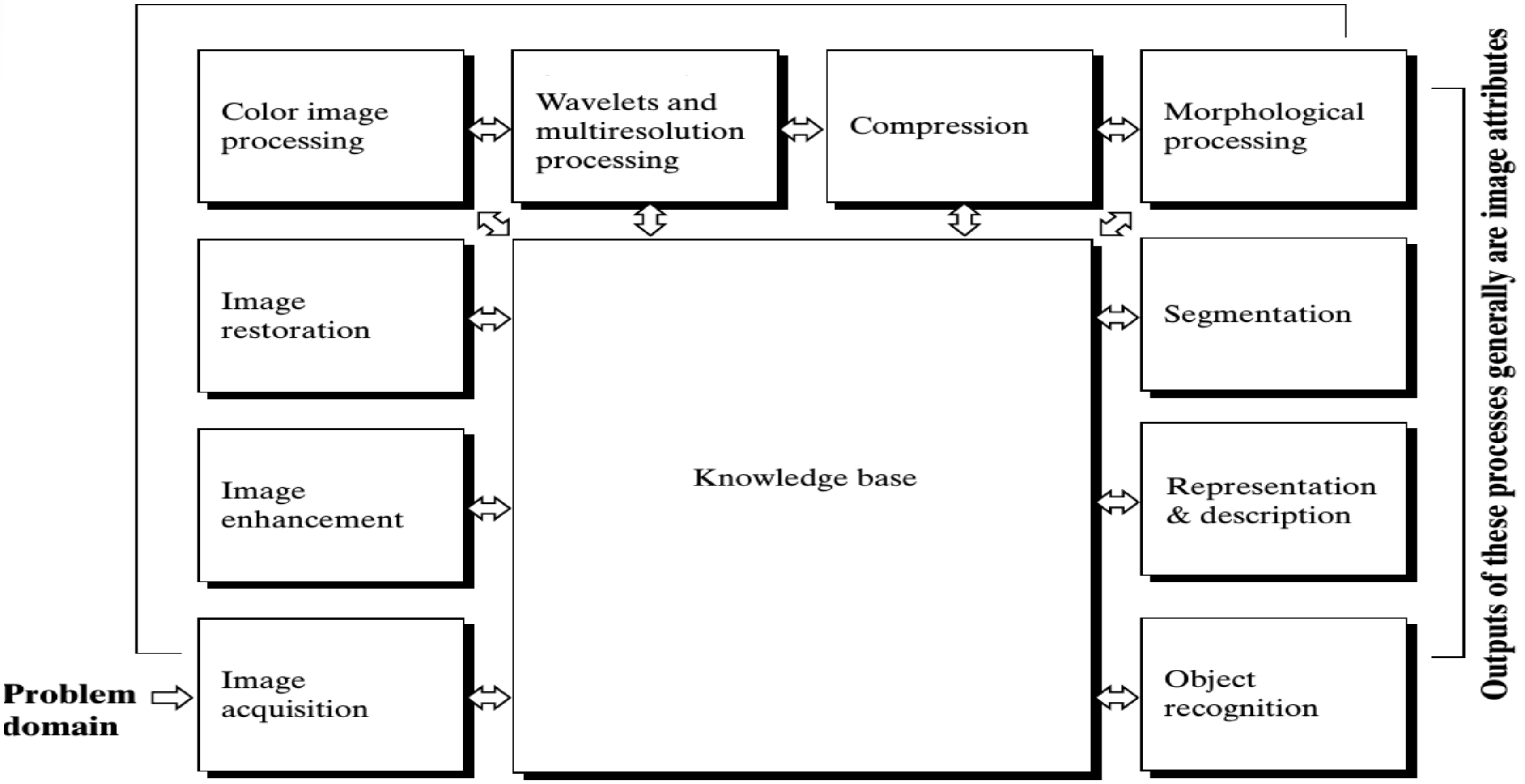


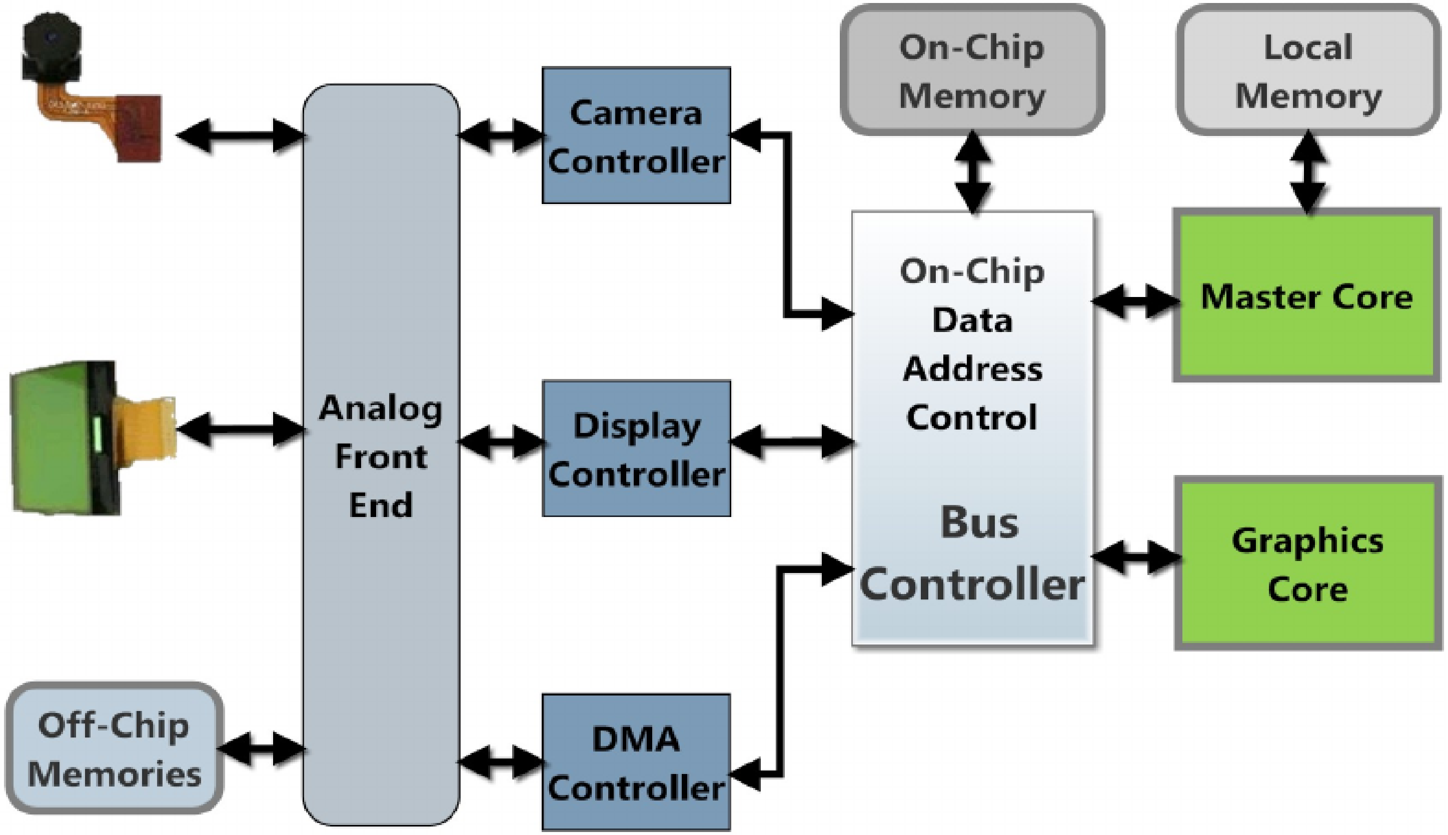
Image Processing Problems



UCERD Gathering Intellectuals www.ucerd.com

BSC Microsoft Research Centre

Graphics System



OpenCV

Intel® OPEN SOURCE COMPUTER VISION LIBRARY

Goals

- Develop a universal toolbox for research and development in the field of Computer Vision

We will talk about:

- Algorithmic content
- Technical content
- Examples of usage
- Trainings

OpenCV algorithms

OpenCV Functionality (more than 350 algorithms)

- Basic structures and operations
- Image Analysis
- Structural Analysis
- Object Recognition
- Motion Analysis and Object Tracking
- 3D Reconstruction

Basic Structures and Operations

- Image and Video Data Structures

Mat image;

Image = imread (“path”);

- Multidimensional array operations

include operations on images, matrices and histograms.

equalizeHist(src, dst);

- Dynamic structures operations

concern all vector data storages.

- Drawing primitives

allows not only to draw primitives but to use the algorithms for pixel access

- Utility functions

in particular, contain fast implementations of useful math functions.

Image Analysis

- Thresholds

  threshold(src_gray, dst, threshold_value, max_BINARY_value, threshold_type);

- Statistics
- Pyramids

  pyrUp and pyrDown (Gaussian and Laplace for sampling)

- Morphology

  Erosion , dilation etc

- Distance transform
- Feature detection

Statistics

- min, max, mean value, standard deviation over the image
- Norms C, L1, L2
- Multidimensional histograms
- Spatial moments up to order 3 (central, normalized, Hu)

Multidimensional Histograms

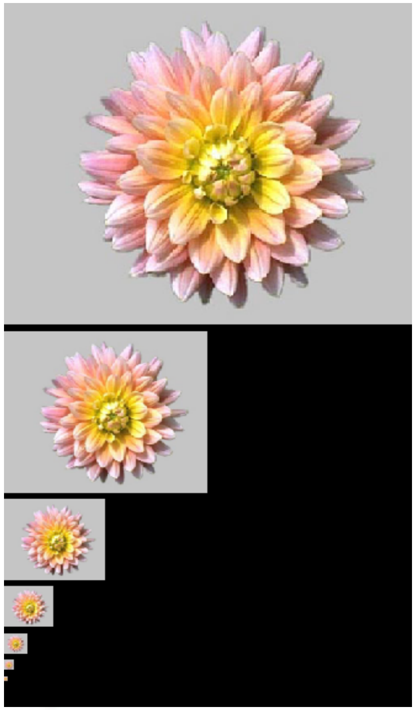
- Histogram operations : calculation, normalization, comparison, back project
- Histograms types:
  - ✓ Dense histograms
  - ✓ Signatures (balanced tree)

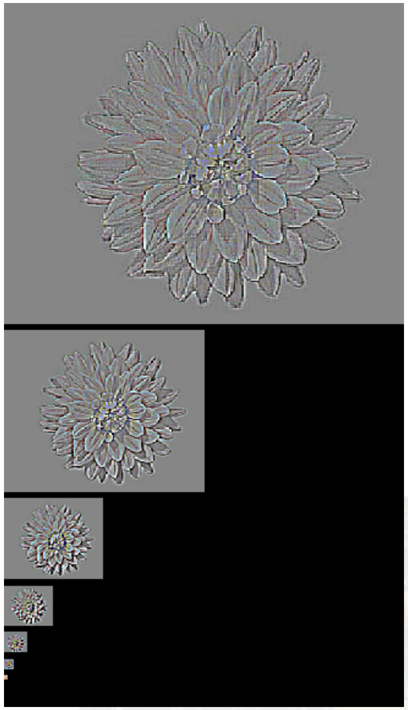
Image Pyramids

- Gaussian and Laplacian pyramids
- Image segmentation by pyramids

Image Pyramids

- Gaussian and Laplacian

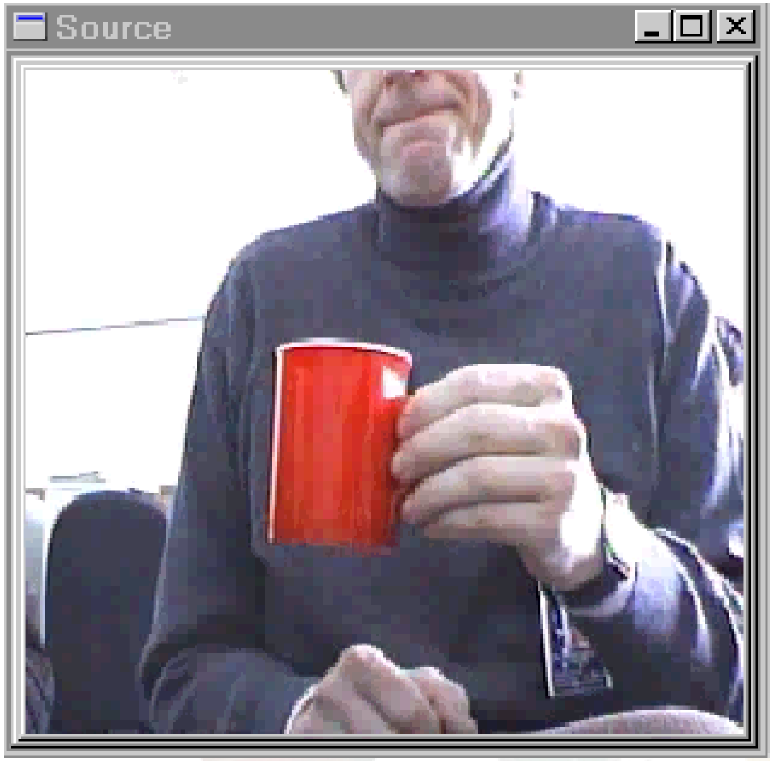


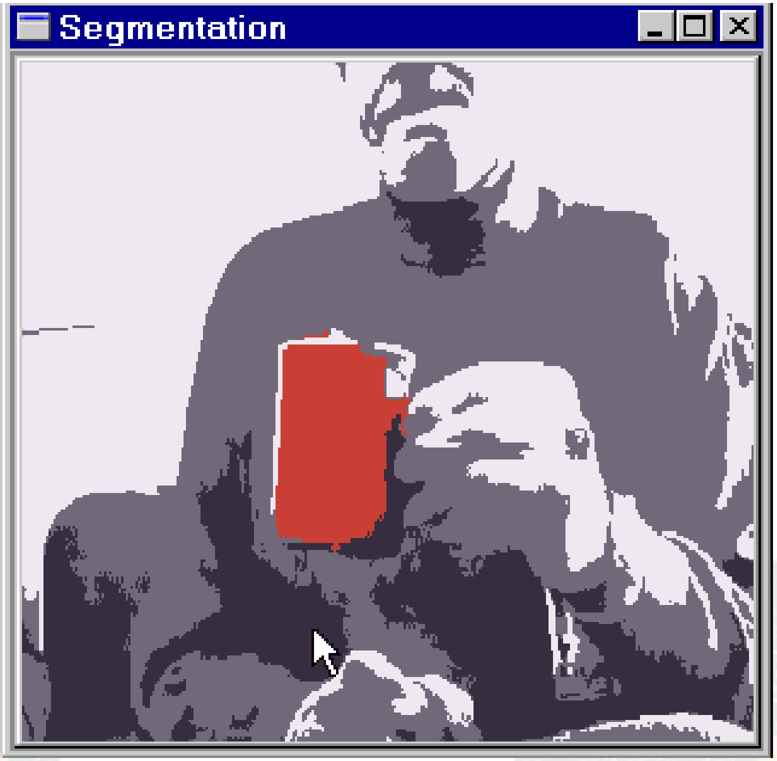


Pyramid-based color segmentation

On still pictures

And on movies



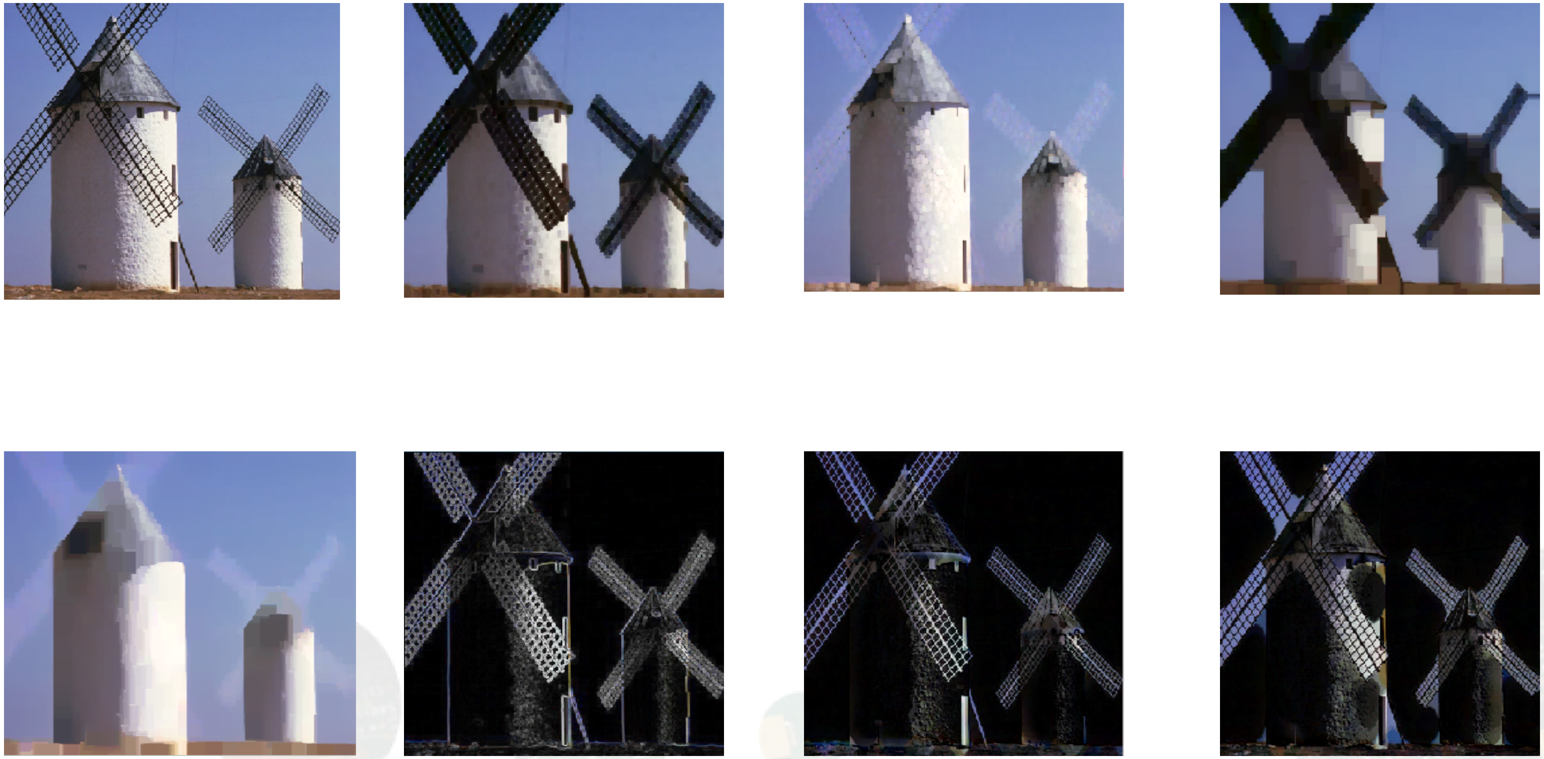


Morphological Operations

- Two basic morphology operations using structuring element:
  - ✓ erosion
  - ✓ dilation
- More complex morphology operations:
  - ✓ opening
  - ✓ closing
  - ✓ morphological gradient
  - ✓ top hat
  - ✓ black hat

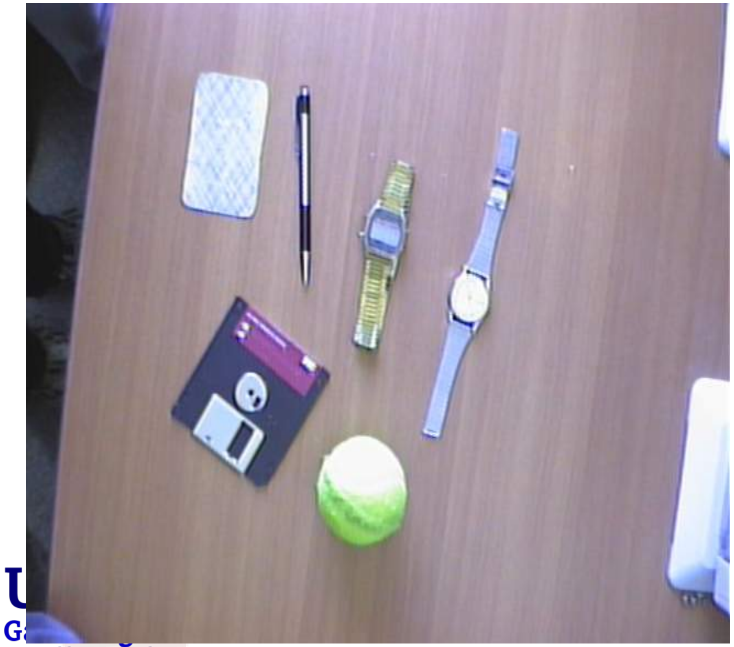
Morphological Operations Examples

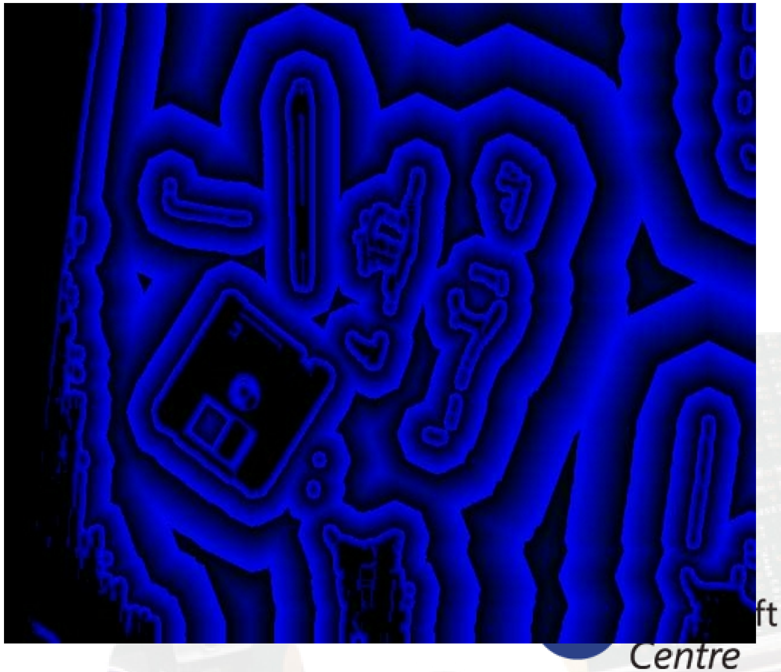
- Morphology - applying Min-Max. Filters and its combinations



Distance Transform

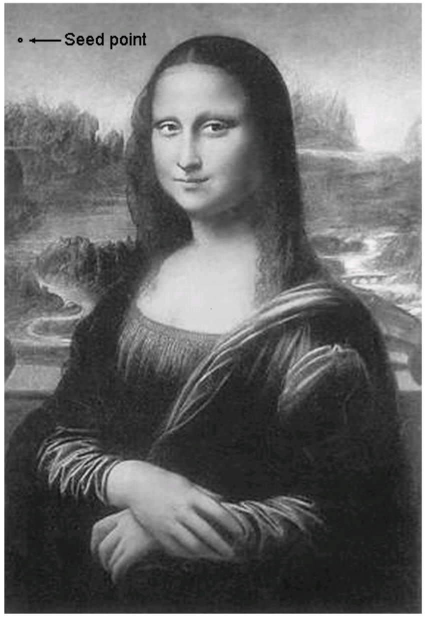
- Calculate the distance for all non-feature points to the closest feature point
- Two-pass algorithm, 3x3 and 5x5 masks, various metrics predefined



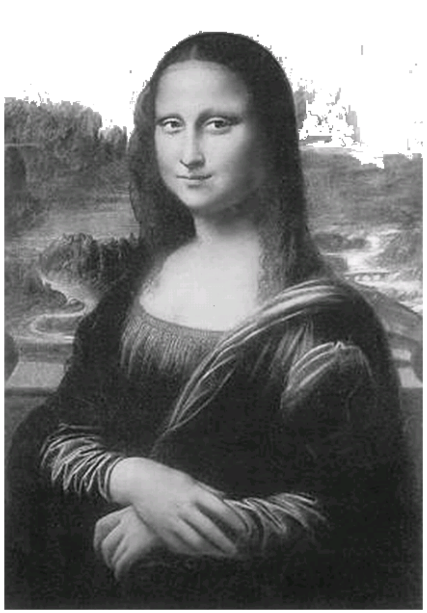


Flood Filling

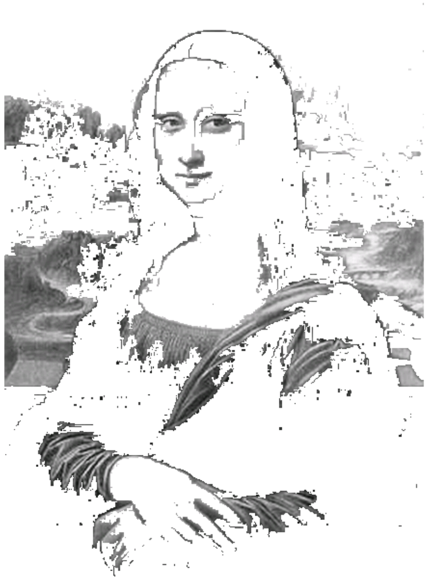
- Simple
- Gradient



Original image



Tolerance interval ± 5

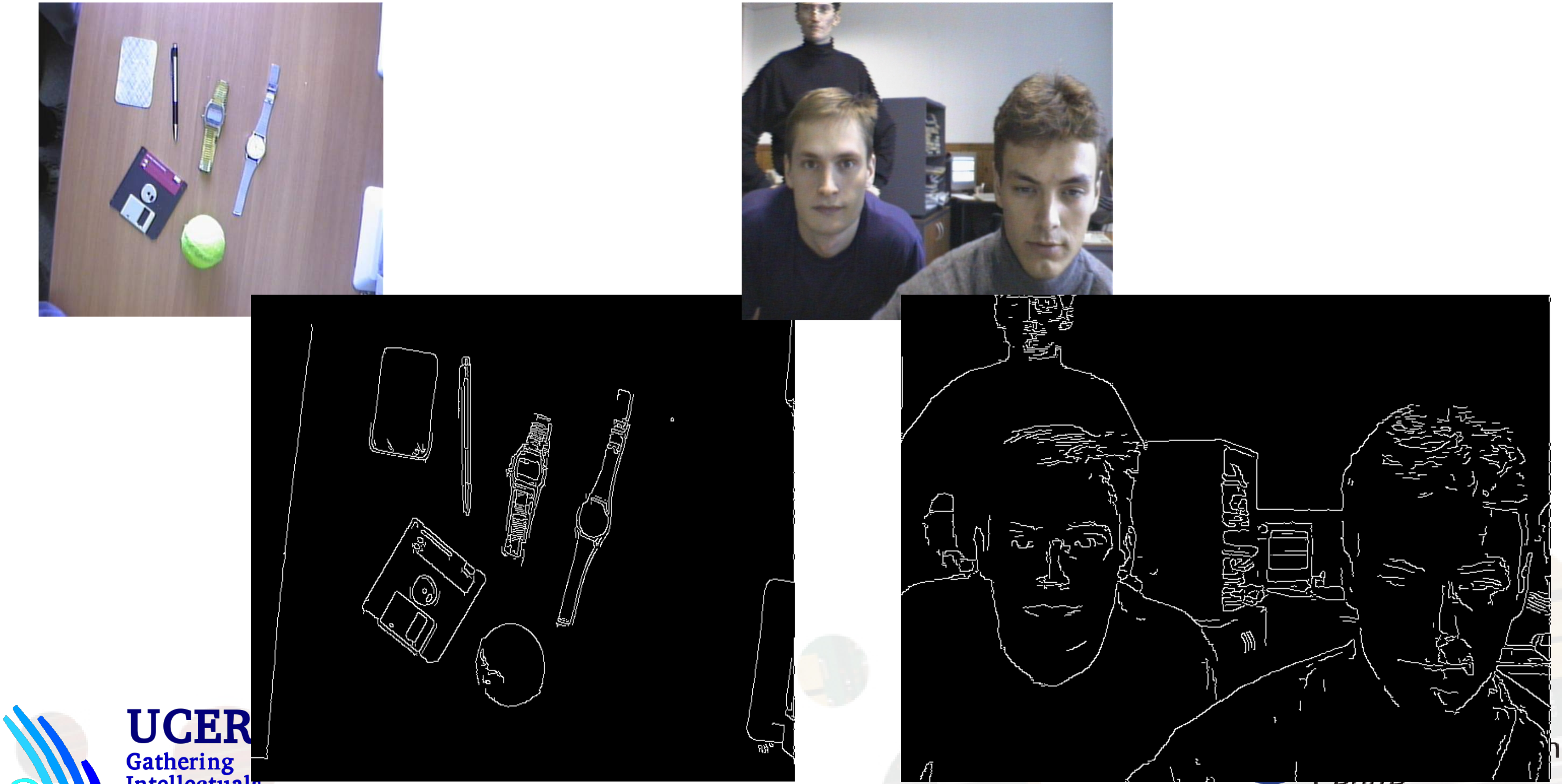


Tolerance interval ± 6

Feature Detection

- Fixed filters (Sobel operator, Laplacian);
- Optimal filter kernels with floating point coefficients (first, second derivatives, Laplacian)
- Special feature detection (corners)
- Canny operator
- Hough transform (find lines and line segments)
- Gradient runs

Canny Edge Detector

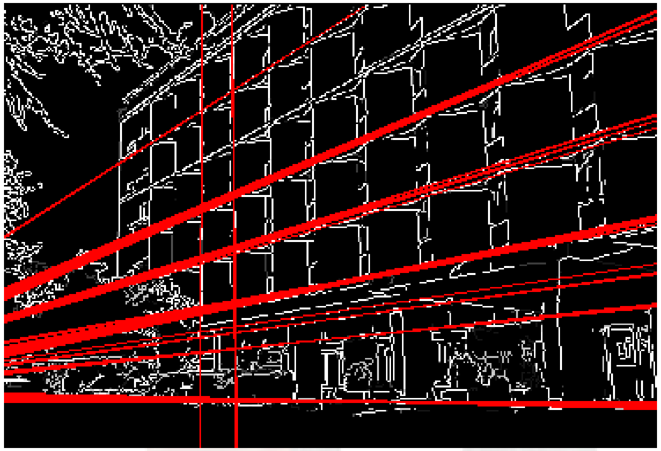


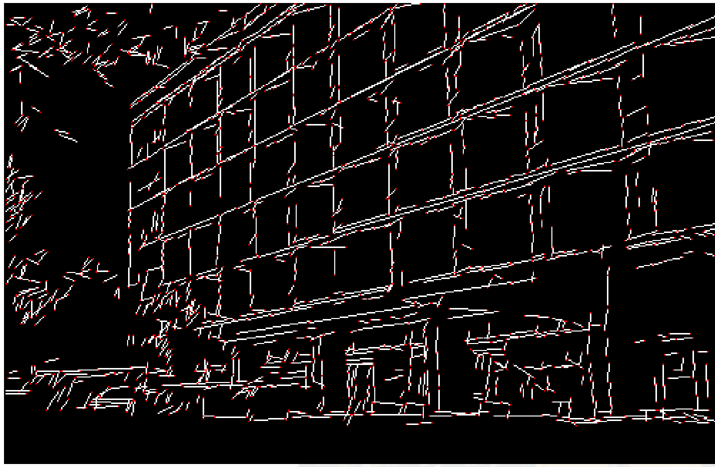
Hough Transform

Detects lines in a binary image

- Standard Hough Transform

- Probabilistic Hough Transform



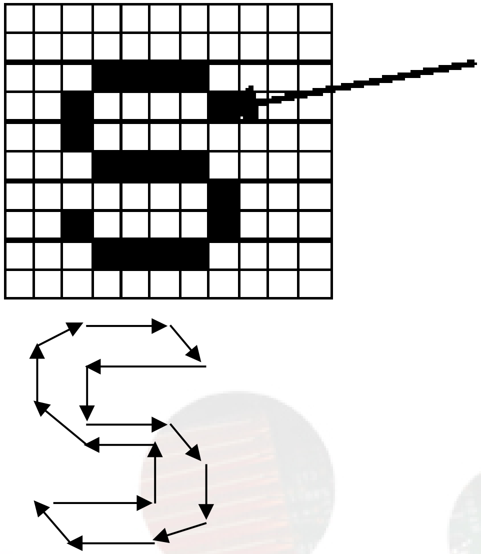


UCERD Gathering Intellectuals www.ucerd.com

BSC Microsoft Research Centre

Contour Retrieving

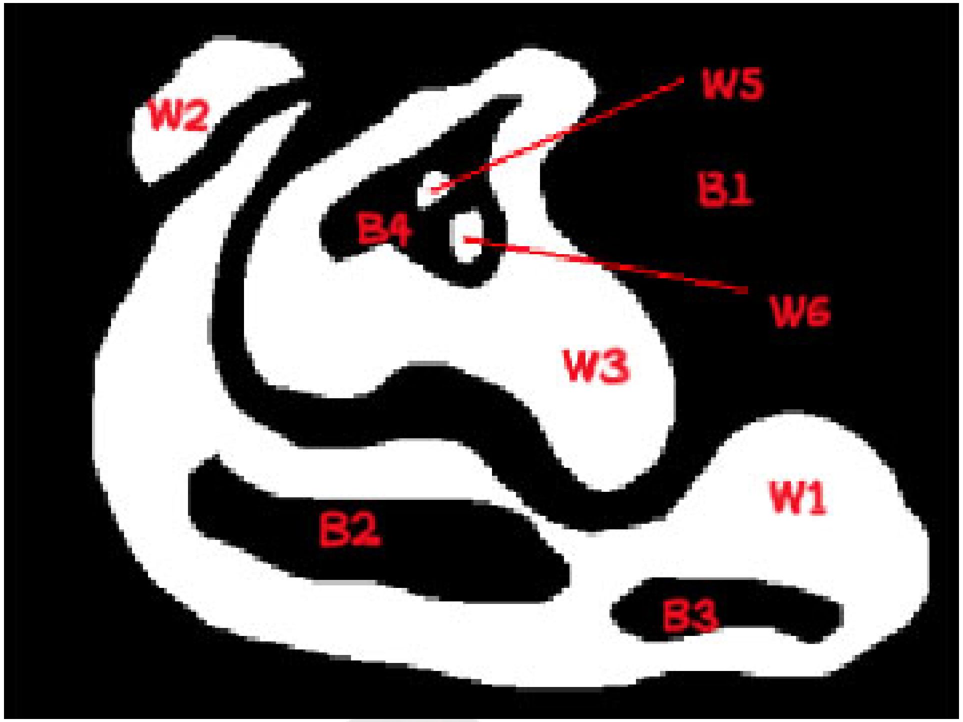
- The contour representation:
  - ✓ Chain code (Freeman code)
  - ✓ Polygonal representation



Initial Point

Chain code for the curve:
34445670007654443

Hierarchical representation of contours



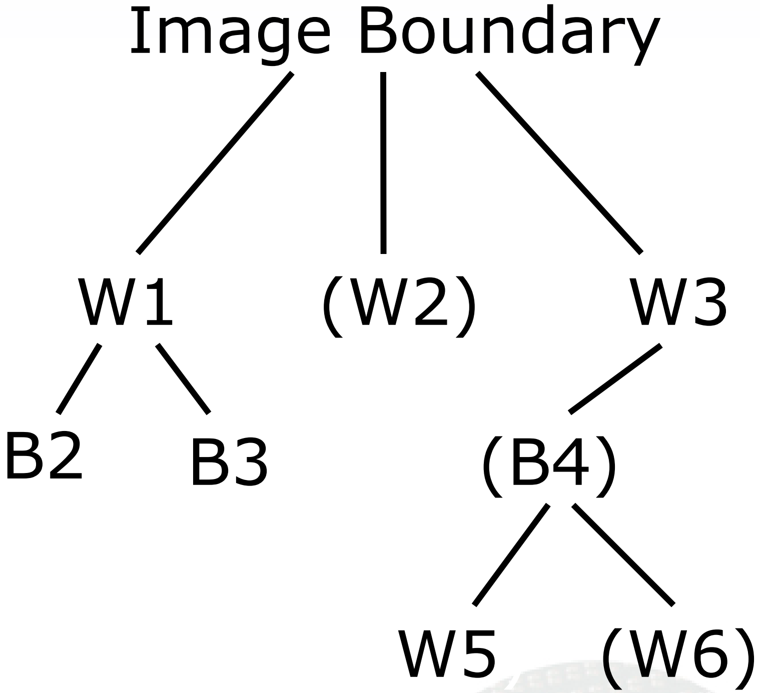


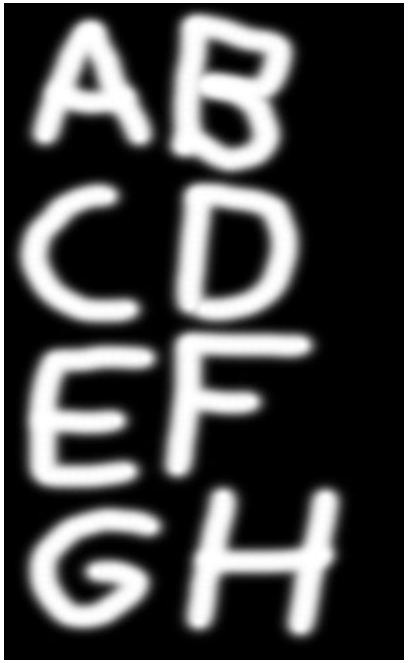
Image Boundary

W1 (W2) W3

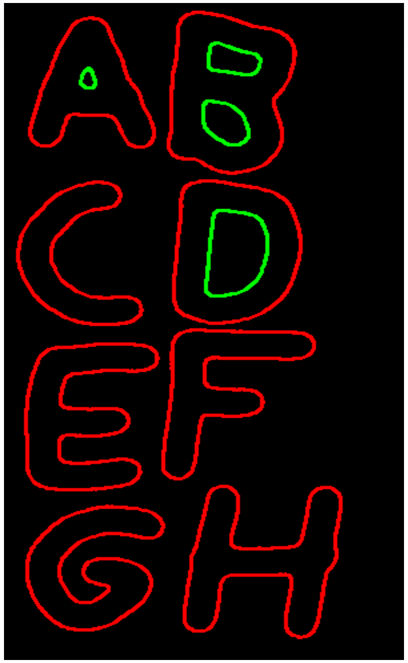
B2 B3 (B4)

W5 (W6)

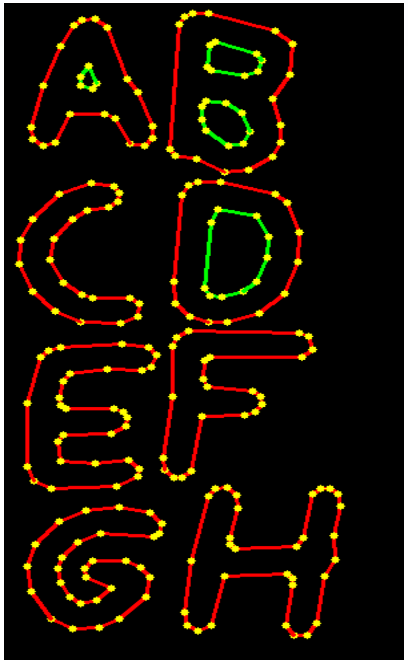
Contours Examples



Source Picture
(300x600 = 180000 pts total)



Retrieved Contours
(<1800 pts total)



After Approximation
(<180 pts total)

And it is rather fast: ~70 FPS for 640x480 on complex scenes

OpenCV Functionality

✓ Basic structures and operations

✓ Image Analysis

- Structural Analysis

■ Object Recognition

■ Motion Analysis and Object Tracking

■ 3D Reconstruction

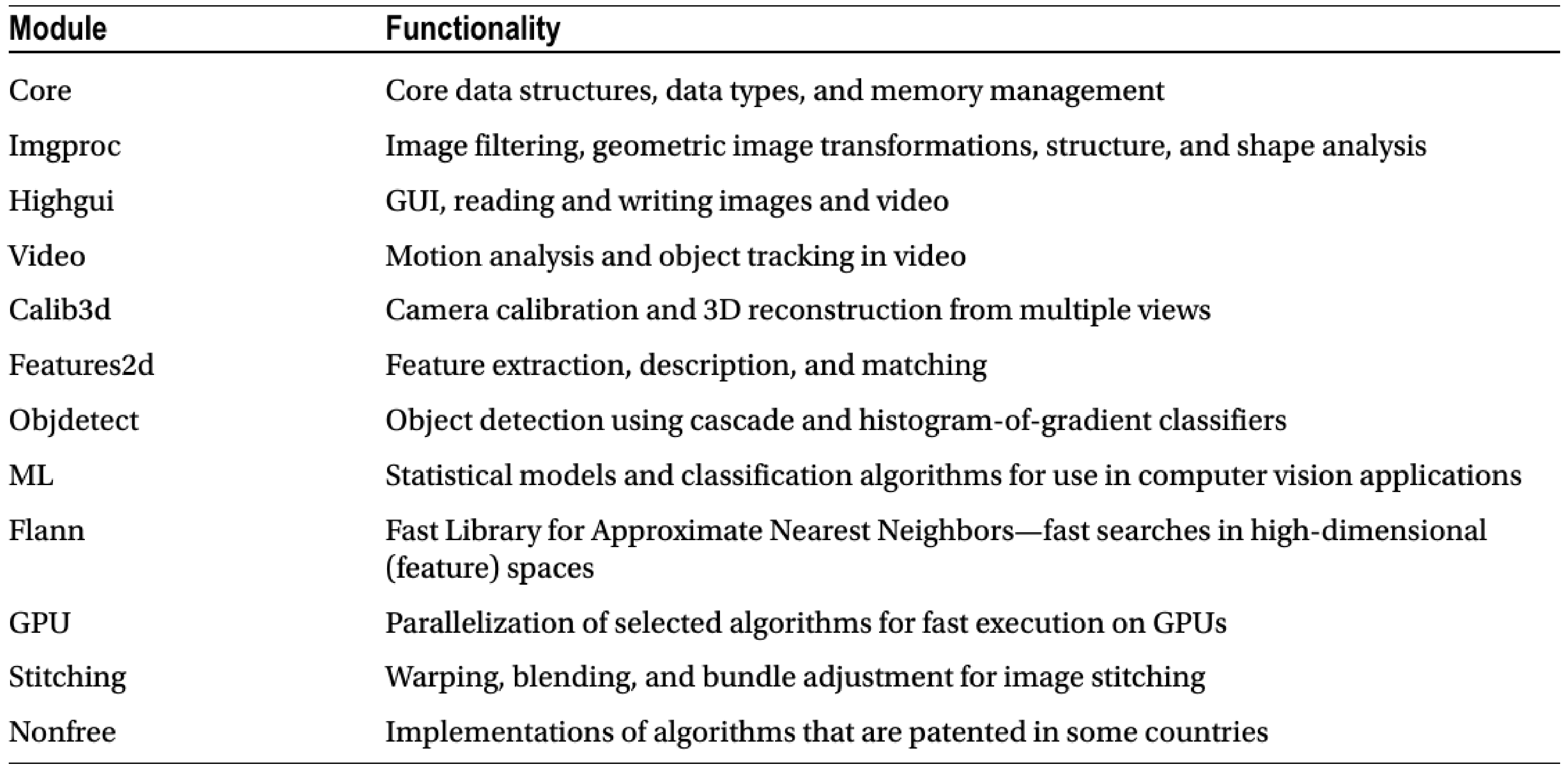
Object Recognition

- Eigen objects
- Hidden Markov Models

We will talk about:

- Algorithmic content
- Technical content
- Examples of usage
- Trainings

OpenCV Modules/Libraries



Module	Functionality
Core	Core data structures, data types, and memory management
Imgproc	Image filtering, geometric image transformations, structure, and shape analysis
Highgui	GUI, reading and writing images and video
Video	Motion analysis and object tracking in video
Calib3d	Camera calibration and 3D reconstruction from multiple views
Features2d	Feature extraction, description, and matching
Objdetect	Object detection using cascade and histogram-of-gradient classifiers
ML	Statistical models and classification algorithms for use in computer vision applications
Flann	Fast Library for Approximate Nearest Neighbors—fast searches in high-dimensional (feature) spaces
GPU	Parallelization of selected algorithms for fast execution on GPUs
Stitching	Warping, blending, and bundle adjustment for image stitching
Nonfree	Implementations of algorithms that are patented in some countries

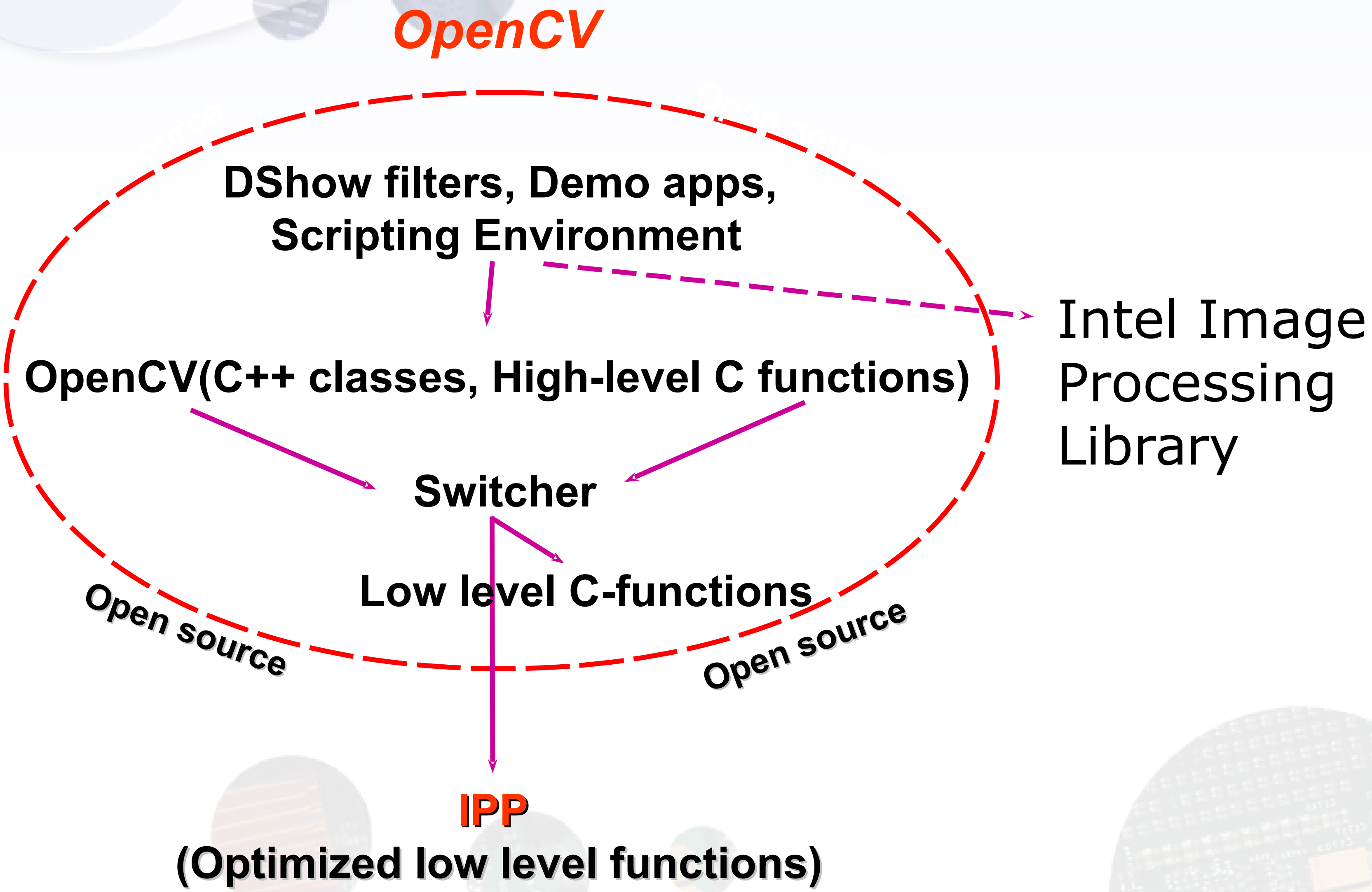
Technical content

- Software requirements
- OpenCV structure
- Data types
- Error Handling
- I/O libraries (HighGUI, CvCAM)
- Scripting
  - Hawk
  - Using OpenCV in MATLAB
- OpenCV lab (code samples)

Software Requirements

- Win32 platforms:
  - **Win9x/WinNT/Win2000**
  - **C++ Compiler** (makefiles for Visual C++ 6.0,Intel C++ Compiler 5.x,Borland C++ 5.5, Mingw GNU C/C++ 2.95.3 are included) for core libraries
  - Visual C++ to build the most of demos
  - DirectX 8.x SDK for directshow filters
  - ActiveTCL 8.3.3 for TCL demos
  - IPL 2.2+ for the core library tests
- Linux/*NIX:
  - C++ Compiler (tested with GNU C/C++ 2.95.x, 2.96, 3.0.x)
  - TCL 8.3.3 + BWidgets for TCL demos
  - Video4Linux + Camera drivers for most of demos
  - IPL 2.2+ for the core library tests

OpenCV structure



OpenCV

DShow filters, Demo apps,
Scripting Environment

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

Switcher

Low level C-functions

Open source

Open source

Intel Image Processing Library

IPP
(Optimized low level functions)

UCERD
Gathering Intellectuals
www.ucerd.com

BSC Microsoft Research Centre

Data Types

- Image (IplImage);
- Matrix (CvMat);
- Histogram (CvHistogram);

Multi-dimensional array

- Dynamic structures (CvSeq, CvSet, CvGraph);
- Spatial moments (CvMoments);
- Helper data types (CvPoint, CvSize, CvTermCriteria, IplConvKernel and others).

Error Handling

- There are no return error codes
- There is a global error status that can be set or checked via special functions
- By default a message box appears if error happens

Portable GUI library (HighGUI)

- Reading/Writing images in several formats (BMP,JPEG,TIFF,PxM,Sun Raster)
- Creating windows and displaying images in it. HighGUI windows remember their content (no need to implement repainting callbacks)
- Simple interaction facilities: trackbars, getting input from keyboard

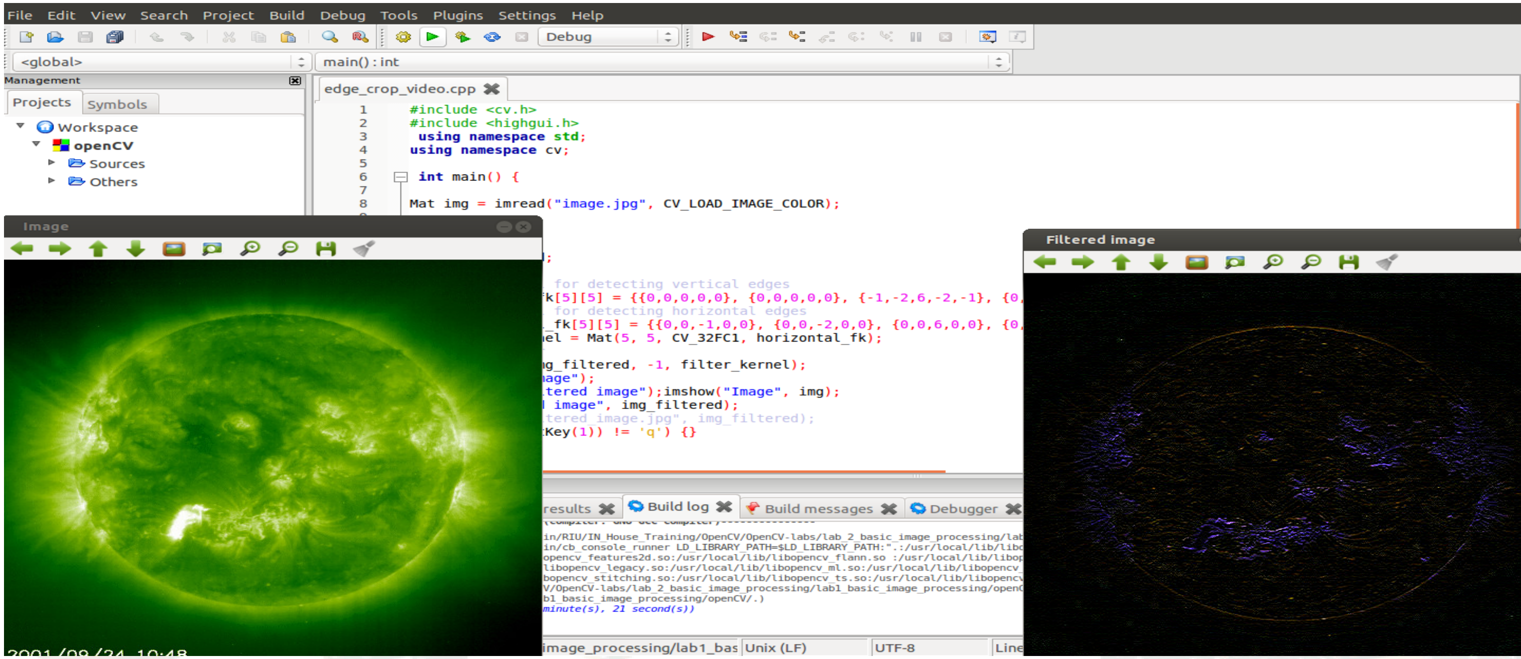
  and mouse (new in Win32 version).

Portable Video Capture Library (CvCAM)

- Single interface for video capture and playback under Linux and Win32
- Provides callback for subsequent processing of frames from camera or AVI-file
- Easy stereo from 2 USB cameras or stereo-camera

ViPS: Visual Processing System

- ARM Multi-core System Architecture
- Visual Environment
- Gnu C/C++ Compiler
- Plugin support
- Interface to OpenCV,IPL and HighGUI via plugins
- Video support



```
#include <cv.h>
#include <highgui.h>
 using namespace std;
using namespace cv;

 int main() {

Mat img = imread("image.jpg", CV_LOAD_IMAGE_COLOR);
```

```
;

 for detecting vertical edges
fk[5][5] = {{0,0,0,0,0}, {0,0,0,0,0}, {-1,-2,6,-2,-1}, {0,
 for detecting horizontal edges
_fk[5][5] = {{0,0,-1,0,0}, {0,0,-2,0,0}, {0,0,6,0,0}, {0,
el = Mat(5, 5, CV_32FC1, horizontal_fk);

g_filtered, -1, filter_kernel);
age");
tered image");imshow("Image", img);
 image", img_filtered);
tered image.jpg", img_filtered);
Key(1)) != 'q') {}
```

```
in/RIU/IN_House_Training/OpenCV/OpenCV-labs/lab_2_basic_image_processing/lab
in/cb_console_runner LD_LIBRARY_PATH=$LD_LIBRARY_PATH:".:/usr/local/lib/libo
opencv_features2d.so:/usr/local/lib/libopencv_flann.so :/usr/local/lib/libop
libopencv_legacy.so:/usr/local/lib/libopencv_ml.so:/usr/local/lib/libopencv_
bopencv_stitching.so:/usr/local/lib/libopencv_ts.so:/usr/local/lib/libopencv
V/OpenCV-labs/lab_2_basic_image_processing/lab1_basic_image_processing/openC
b1_basic_image_processing/openCV/.)
minute(s), 21 second(s))
```

image_processing/lab1_bas | Unix (LF) | UTF-8 | Line

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

Trainings

Start Lab