BSC~Microsoft Research Centre

Introduction to ComputerVision

Tassadaq Hussain

Associate Prof: Riphah Int'l University

Microsoft Barcelona Supercomputing Center

Universitat Politécnica de Catalunya

Barcelona, Spain

UCERD
Gathering Intellectuals
www.ucerd.com

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 Read/Write

- Import cv2 as cv

```
gray_img = cv2.imread('images/input.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('Grayscale', gray_img)
cv2.waitKey()

cv2.imwrite('images/output.jpg', gray_img)

gray_img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
yuv_img = cv2.cvtColor(img, cv2.COLOR_BGR2YUV)
```

Image Analysis

- Thresholds

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

- Statistics
- Pyramids
- 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)

Pyramids

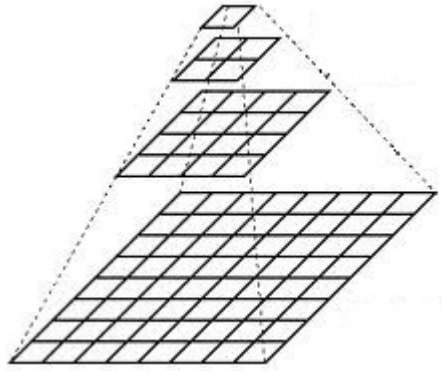
An image pyramid is a collection of images - all arising from a single original image - that are successively downsampled until some desired stopping point is reached.

PyrUp()

pyrdown()

Gaussian pyramid:

Laplacian pyramid:



UCERD
Gathering Intellectuals
www.ucerd.com

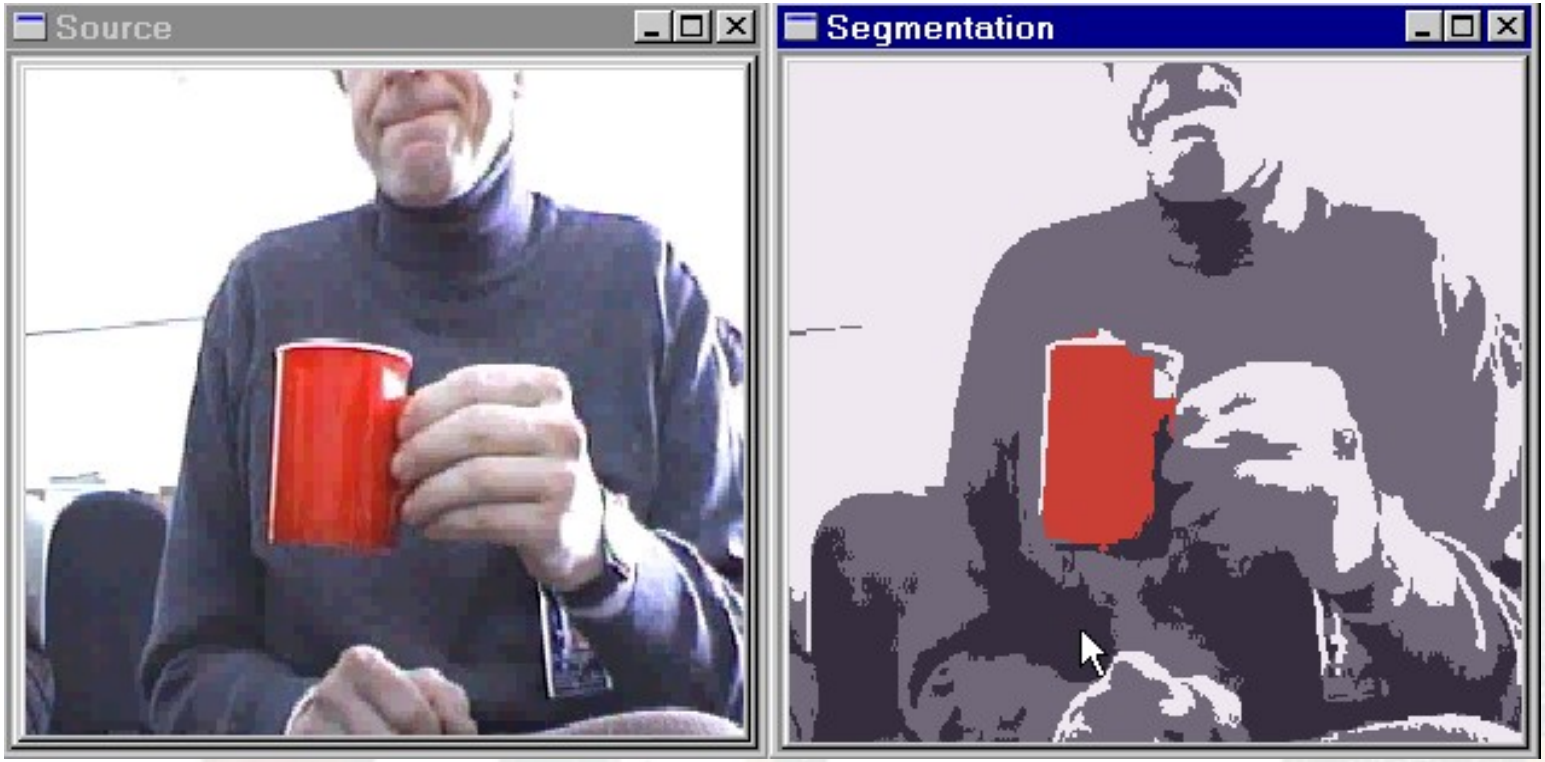
BSC~Microsoft Research Centre

Image Pyramids

- Gaussian and Laplacian

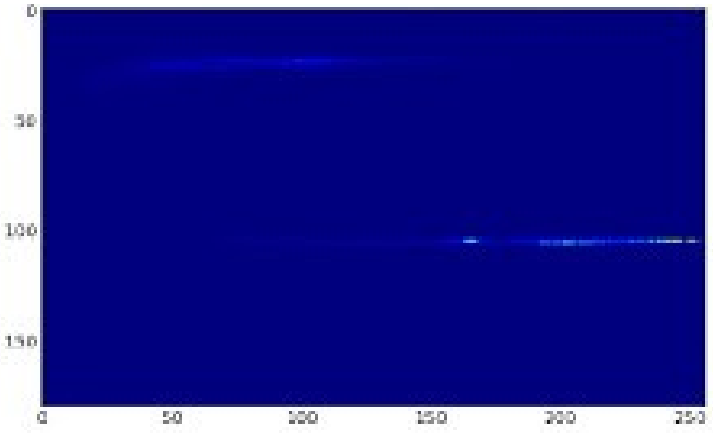
Pyramid-based color segmentation

On still pictures ***And on movies***



Multidimensional Histograms

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



UCERD
Gathering Intellectuals
www.ucerd.com

BSC Microsoft Research Centre

Morphological Operations

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

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~ Microsoft Research Centre

Morphological Operations Examples

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

Image I

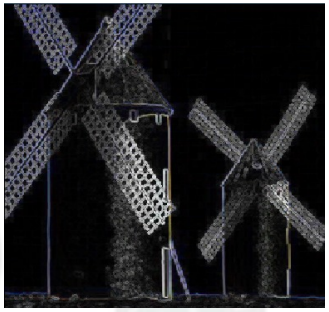
Erosion IΘB

Dilatation I⊕B

Opening IoB= (IΘB)⊕B

Closing I•B= (I⊕B)ΘB

Grad(I)= (I⊕B)-(IΘB)



TopHat(I)= I - (IΘB)

BlackHat(I)= (I⊕B) - I

UCERD
Gathering Intellectuals
www.ucerd.com

BSC Microsoft Research Centre

Distance Transform

The distance transform operator generally takes binary images as inputs. In this operation, the gray level intensities of the points inside the foreground regions are changed to distance their respective distances from the closest 0 value (boundary).

distanceTransform()

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

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

Flood Filling

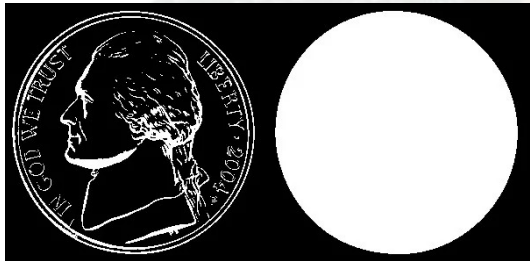
- Simple
- Gradient

```
cv2.floodFill(img, mask, (0,0), 255);
```

Original image

Tolerance interval ± 5

Tolerance interval ± 6



Feature Detection

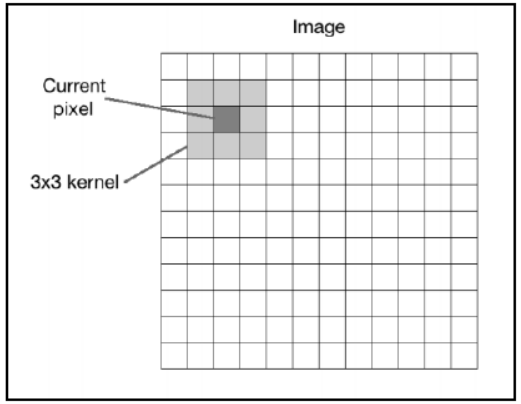
- Fixed filters (Sobel operator, Laplacian);
- Optimal filter kernels with floating point coefficients (first, second derivatives, Laplacian)
- Special feature detection (corners)
- Canny operator
- Hough transform (find lines and line segments)
- Gradient runs

Convolution

Convolution is a fundamental operation in image processing. It basically applies a mathematical operator to each pixel, and change its value in some way.

To apply this mathematical operator, convolution uses another matrix called a kernel. The kernel is usually much smaller in size than the input image. For each pixel in the image, we take the kernel and place it on top so that the center of the kernel coincides with the pixel under consideration.

We then multiply each value in the kernel matrix with the corresponding values in the image, and then sum it up. This is the new value that will be applied to this position in the output image.



UCERD
Gathering Intellectuals
www.ucerd.com

```
import cv2
import numpy as np
img = cv2.imread('images/input.jpg')
rows, cols = img.shape[:2]
kernel_identity = np.array([[0,0,0], [0,1,0], [0,0,0]])
kernel_3x3 = np.ones((3,3), np.float32) / 9.0 # Divide by 9 to normalize the kernel
kernel_5x5 = np.ones((5,5), np.float32) / 25.0 # Divide by 25 to normalize the kernel
cv2.imshow('Original', img)
# value -1 is to maintain source image depth
output = cv2.filter2D(img, -1, kernel_identity)
cv2.imshow('Identity filter', output)
output = cv2.filter2D(img, -1, kernel_3x3)
cv2.imshow('3x3 filter', output)
output = cv2.filter2D(img, -1, kernel_5x5)
cv2.imshow('5x5 filter', output)
cv2.waitKey(0)
```

UCERD
Gathering Intellectuals
www.ucerd.com

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
img = cv2.imread('images/input.jpg')
cv2.imshow('Original', img)
size = 15
# generating the kernel
kernel_motion_blur = np.zeros((size, size))
kernel_motion_blur[int((size-1)/2), :] = np.ones(size)
kernel_motion_blur = kernel_motion_blur / size
# applying the kernel to the input image
output = cv2.filter2D(img, -1, kernel_motion_blur)
cv2.imshow('Motion Blur', output)
cv2.waitKey(0)
```

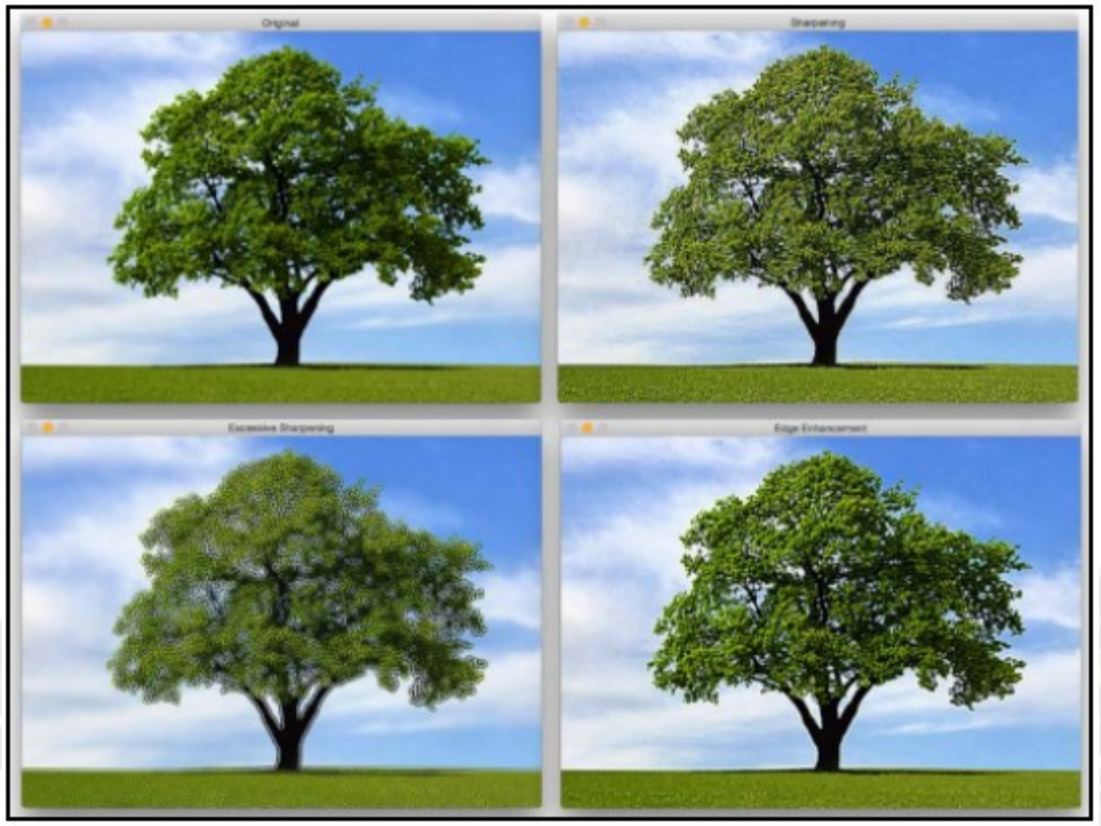
UCERD
Gathering Intellectuals
www.ucerd.com

Sharpening Images

```
# generating the kernels
kernel_sharpen_1 = np.array([[-1,-1,-1], [-1,9,-1], [-1,-1,-1]])
kernel_sharpen_2  = np.array([[1,1,1], [1,-7,1], [1,1,1]])
kernel_sharpen_3  = np.array([[-1,-1,-1,-1,-1], [-1,2,2,2,-1], [-1,2,8,2,-1], [-1,2,2,2,-1], [-1,-1,-1,-1,-1]]) / 8.0
```

$$M = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -7 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



The process of edge detection involves detecting sharp edges in the image, and producing a binary image as the output. Typically, we draw white lines on a black background to indicate those edges.

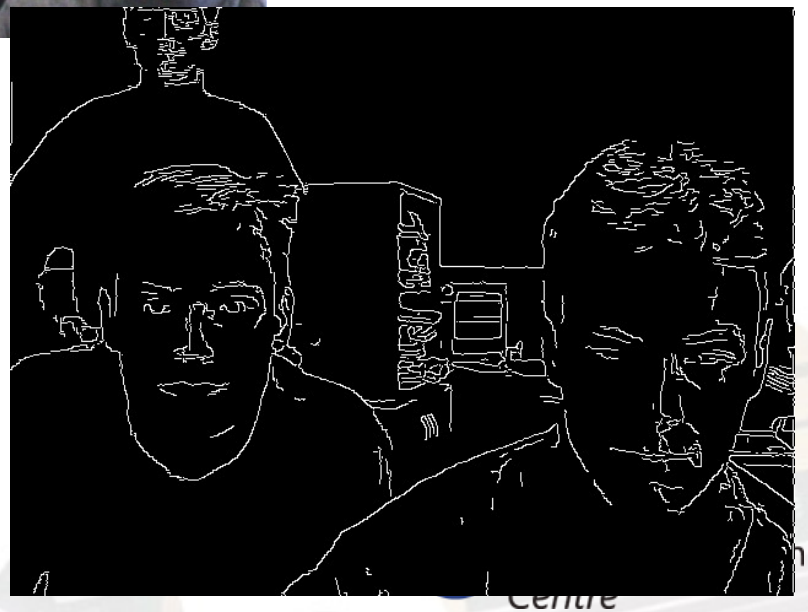
We can think of edge detection as a high pass filtering operation. A high pass filter allows high-frequency content to pass through and blocks the low-frequency content. As we discussed earlier, edges are high-frequency content. In edge detection, we want to retain these edges and discard everything else. Hence, we should build a kernel that is the equivalent of a high pass filter.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

UCERD
Gathering Intellectuals
www.ucerd.com

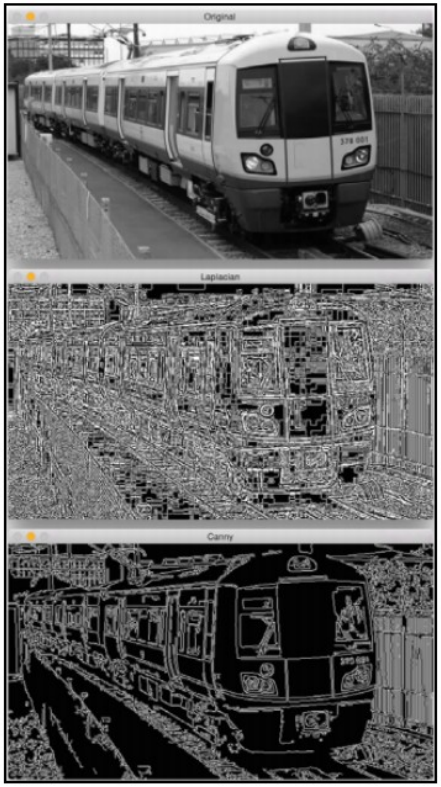
BSC Microsoft Research Centre

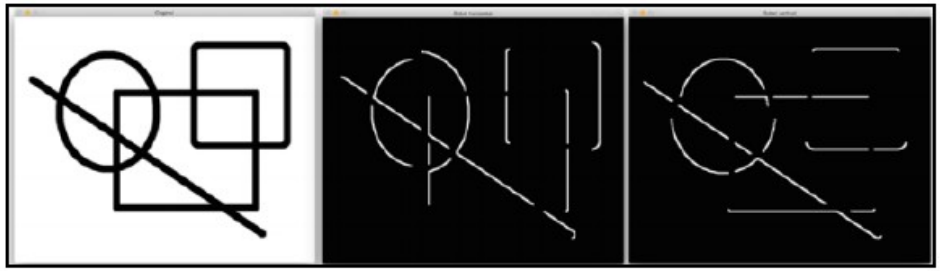
Canny Edge Detector



UCER
Gathering
Intellectuals
www.ucerd.com

```
import cv2
import numpy as np
img = cv2.imread('images/input_shapes.png',
cv2.IMREAD_GRAYSCALE)
rows, cols = img.shape # It is used depth of
cv2.CV_64F.
sobel_horizontal = cv2.Sobel(img, cv2.CV_64F, 1, 0,
ksize=5)
# Kernel size can be: 1,3,5 or 7.
sobel_vertical = cv2.Sobel(img, cv2.CV_64F, 0, 1,
ksize=5)
cv2.imshow('Original', img)
cv2.imshow('Sobel horizontal', sobel_horizontal)
cv2.imshow('Sobel vertical', sobel_vertical)
cv2.waitKey(0)
```





UCERD
Gathering Intellectuals
www.ucerd.com

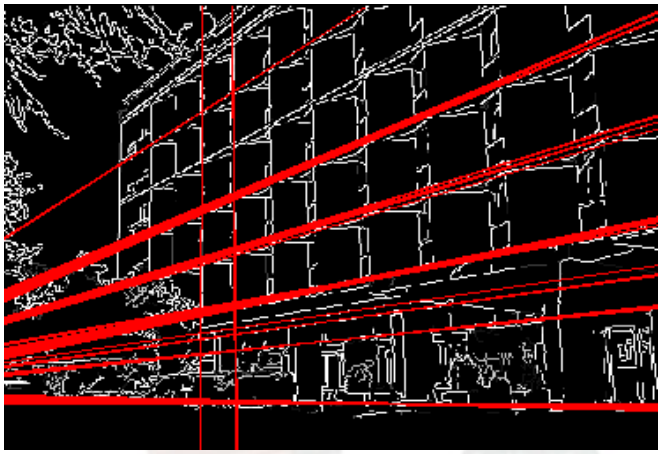
BSC Microsoft Research Centre

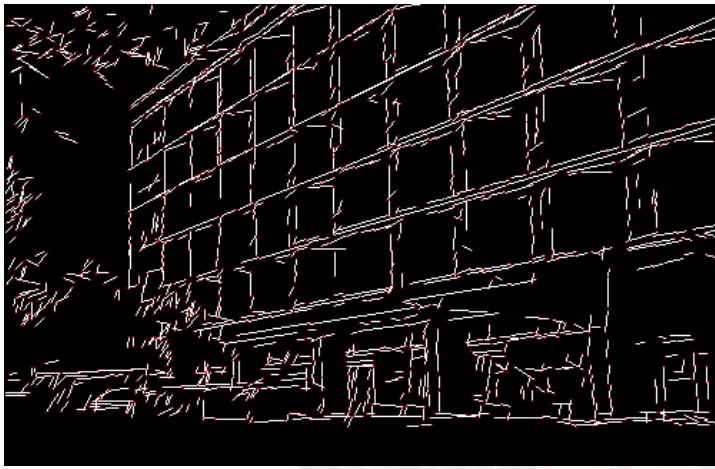
Hough Transform

Detects lines in a binary image

- Standard Hough Transform

- Probabilistic Hough Transform





Hough Transform

Detects lines in a binary image

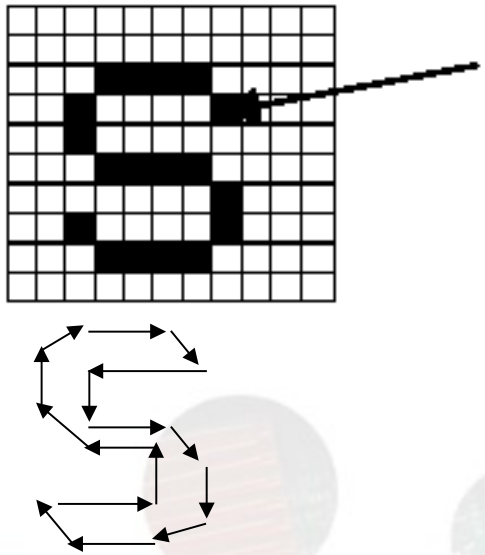
Hough Transform is a popular technique to detect any shape, if you can represent that shape in mathematical form. It can detect the shape even if it is broken or distorted a little bit. We will see how it works for a line.

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

Contour Retrieving

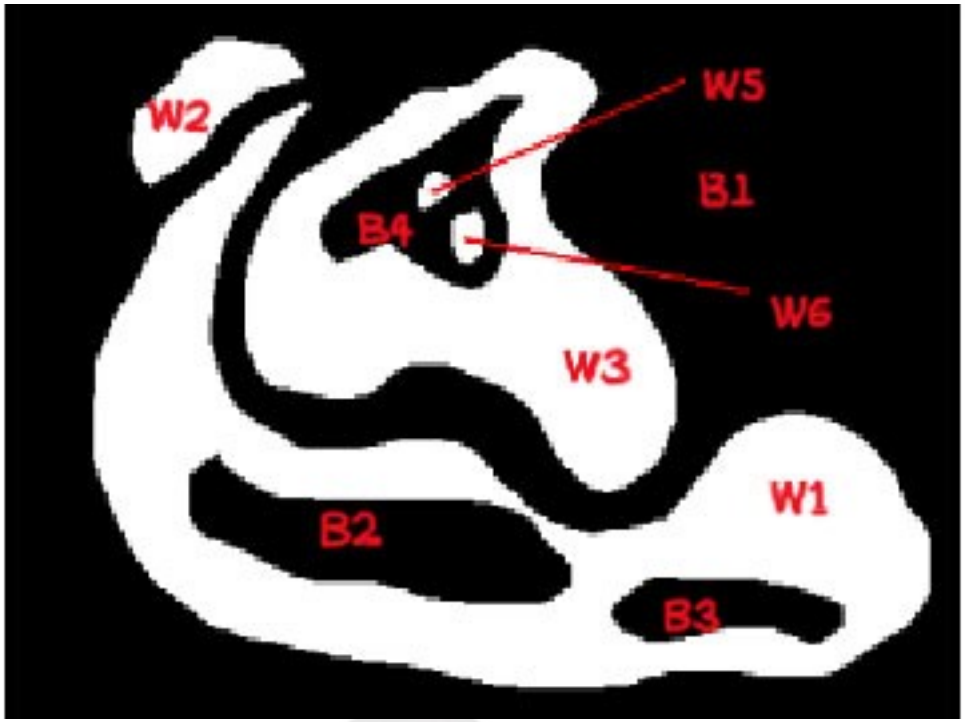
- The contour representation:
  - ✓ Chain code (Freeman code)
  - ✓ Polygonal representation

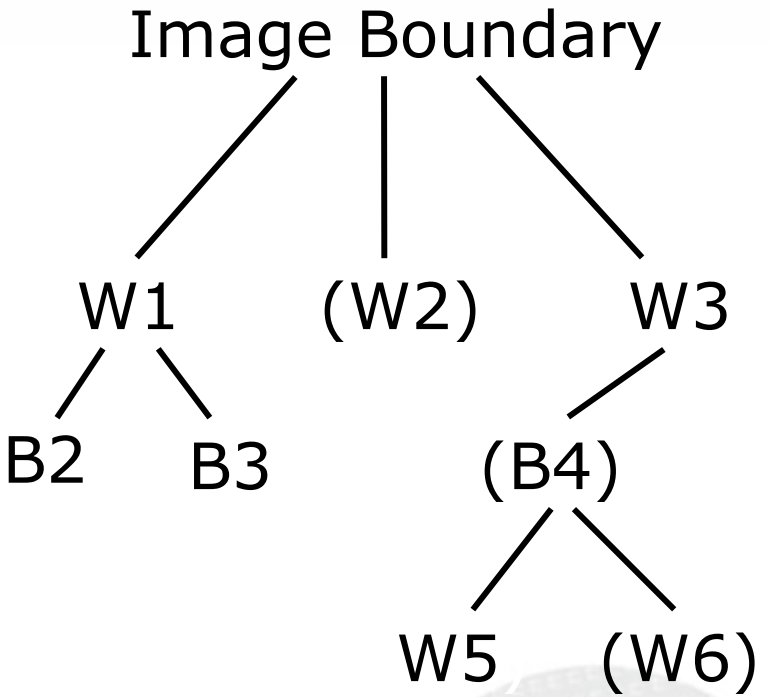


Initial Point

Chain code for the curve:
3444567000765443

Hierarchical representation of contours





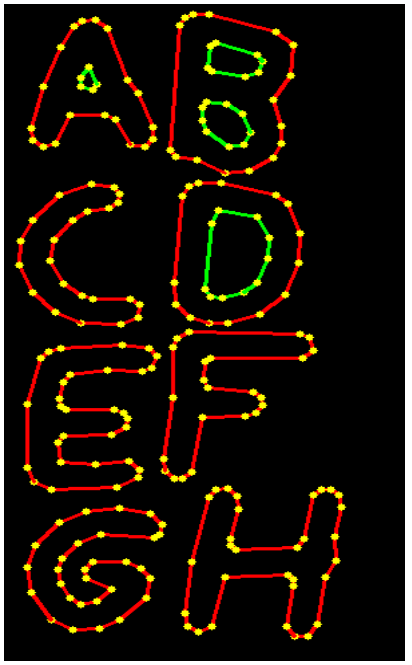
UCERD
Gathering Intellectuals
www.ucerd.com

BSC Microsoft Research Centre

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

UCERD
Gathering Intellectuals
www.ucerd.com

BSC Microsoft Research Centre

OpenCV Functionality

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

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~ Microsoft Research Centre

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

UCERD
Gathering Intellectuals
www.ucerd.com
BSC~Microsoft Research Centre

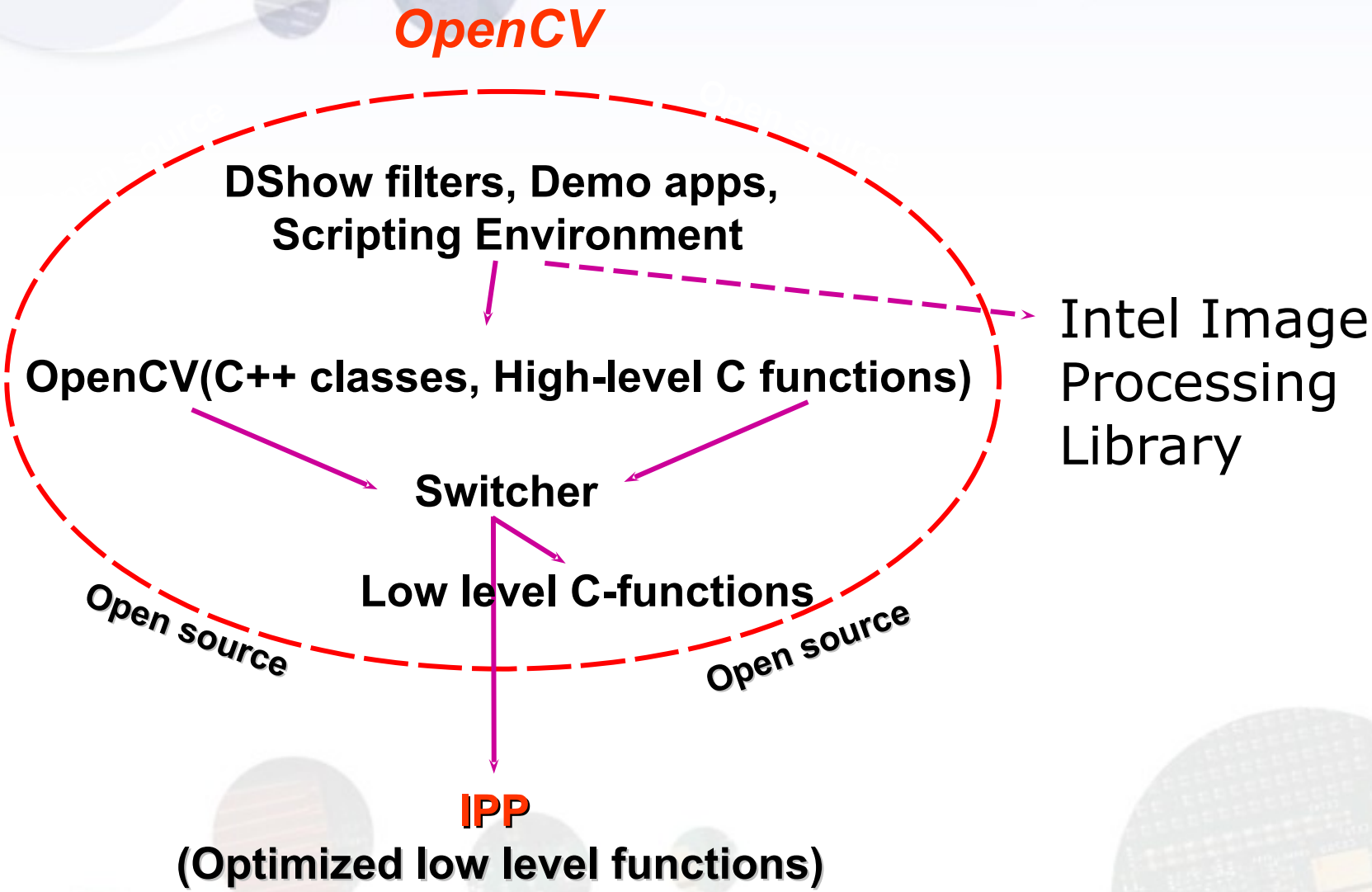
Technical content

- 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



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).

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

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).

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

Portable Video Capture Library (CvCAM)

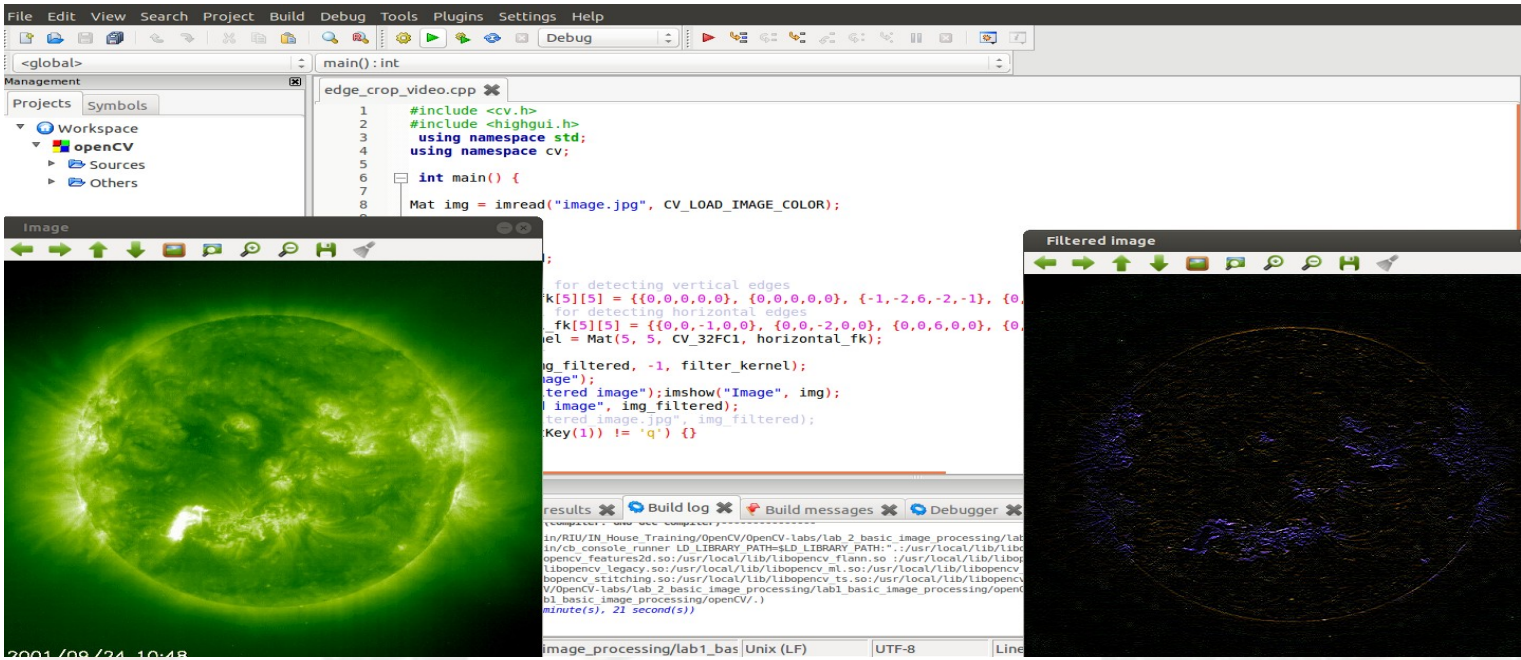
- Single interface for video capture and playback under Linux and Win32
- Provides callback for subsequent processing of frames from camera or AVI-file
- Easy stereo from 2 USB cameras or stereo-camera

UCERD
Gathering Intellectuals
www.ucerd.com

BSC~ Microsoft Research Centre

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



UCERD
Gathering Intellectuals
www.ucerd.com

BSC~Microsoft Research Centre

Trainings

Start Lab