ME336 Collaborative Robot Learning

Song Chaoyang

Assistant Professor

Department of Mechanical and Energy Engineering

songcy@sustc.edu.cn

Agenda

Week 07, Wed, DATE

- Image processing: OpenCV
  - OpenCV-Python
- Point cloud processing: PCL

AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

Open Source Computer Vision

- OpenCV is one of the popular open source real time computer vision libraries.
- OpenCV can be programmed using C/C++, Python, and Java
- Installation with ros: The ROS perception metapackage of ROS containing all the perception-related packages, such as OpenCV, PCL, and so on.
- $sudo apt-get install ros-kinetic-perception

Ref:
Mastering ROS for Robotics Programming. Chapter 8.
http://opencv.org/,
https://aur.archlinux.org/packages/ros-kinetic-perception/

AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

General Image Processing Functions

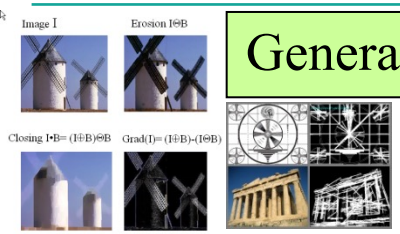
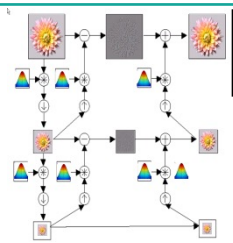
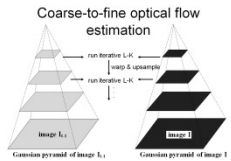
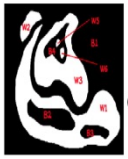


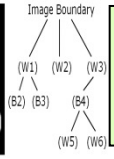
Image Pyramids

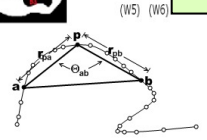


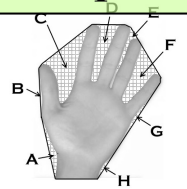


Geometric Descriptors

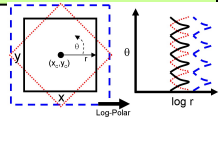




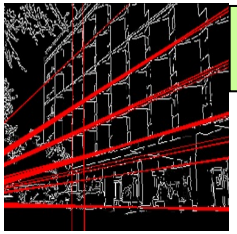


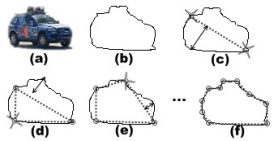


Segmentation

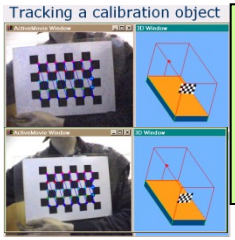


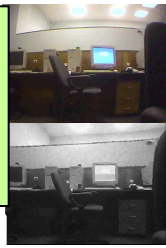
Features



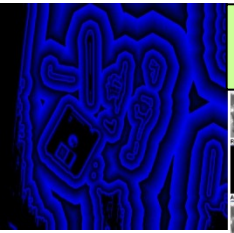


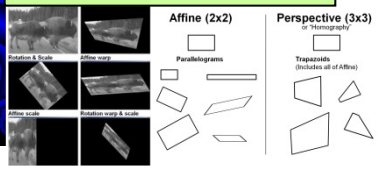
Camera Calibration, Stereo, 3D



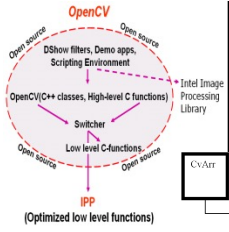


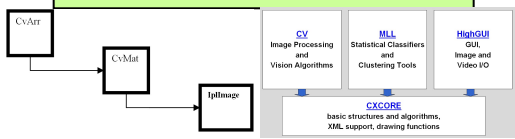
Transforms





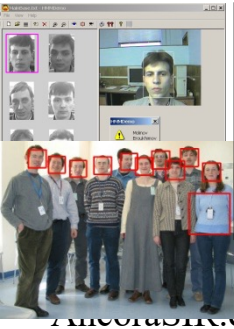
Utilities and Data Structures

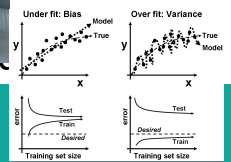




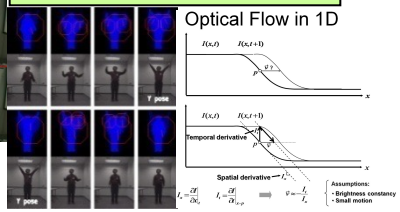
Machine Learning:

- Detection,
- Recognition

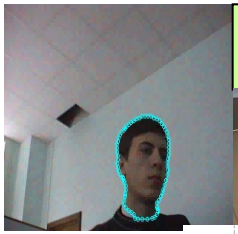


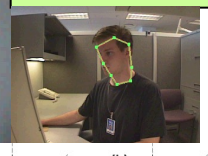


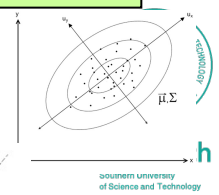
Tracking



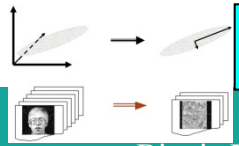
Fitting

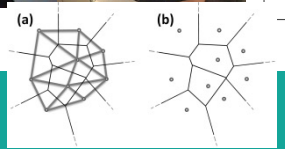






Matrix Math





OpenCV

OpenCV-Python

- We are going to use OpenCV-Python API for our projects.
- A Python wrapper around original OpenCV's C++ implementation.
- An appropriate tool for fast prototyping of computer vision problems: OpenCV combining with Numpy, Scipy, Matplotlib
- Reference:
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html

AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

Highlight

- Basic Operations on Images
- Image Thresholding
- Canny Edge Detection
- Contours in OpenCV
- Template Matching
- Image Segmentation with Watershed Algorithm

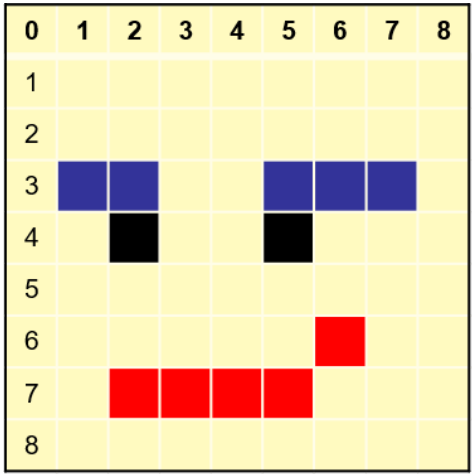
AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

Basic Operations on Images

- Data types: All the OpenCV array structures are converted to-and-from Numpy arrays.
- Reading/Display image: Color images returns an array of Blue, Green, Red values.




```
>>> import cv2
>>> import numpy as np
>>> img = cv2.imread('messi5.jpg')
>>> cv2.imshow(img)
# accessing only blue pixel
>>> blue = img[100,100,0]
>>> print blue
157
# modifying pixel value
>>> img[100,100] = [255,255,255]
```

AncoraSIR.com

SUSTech Southern University of Science and Technology

OpenCV

Basic Operations on Images

- Accessing image properties:

```
>>> print img.shape
(342, 548, 3)
>>> print img.dtype
Uint8
```

- Image ROI(Region of Interest):

```
>>> ball = img[280:340, 330:390]
```

- Splitting/Merging Image Channels

```
>>> b,g,r = cv2.split(img)
>>> img = cv2.merge((b,g,r))
```

AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

Image Thresholding

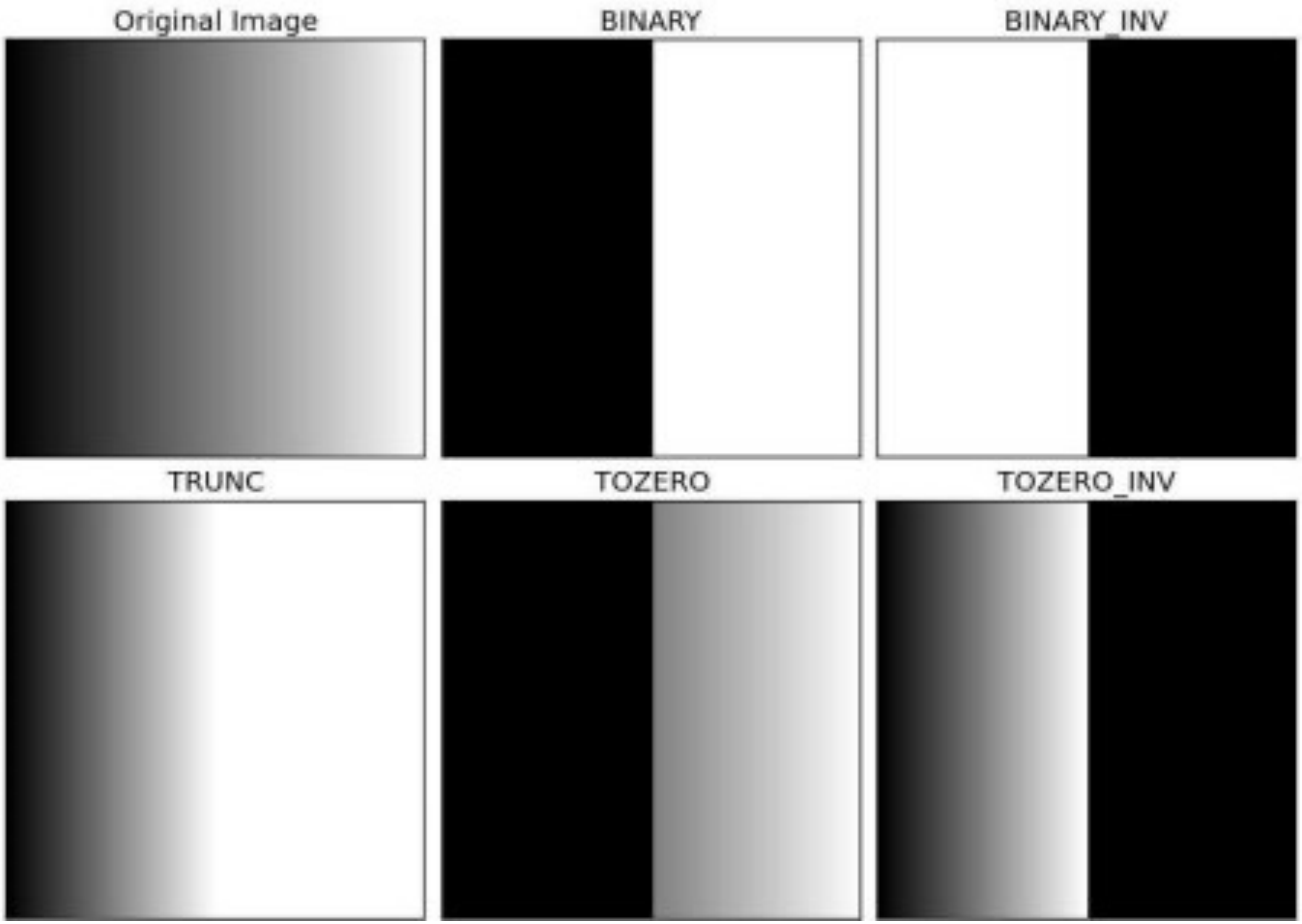
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gradient.png',0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)
titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

for i in xrange(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])

plt.show()
```

AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

Image Thresholding



AncoraSIR.com

OpenCV

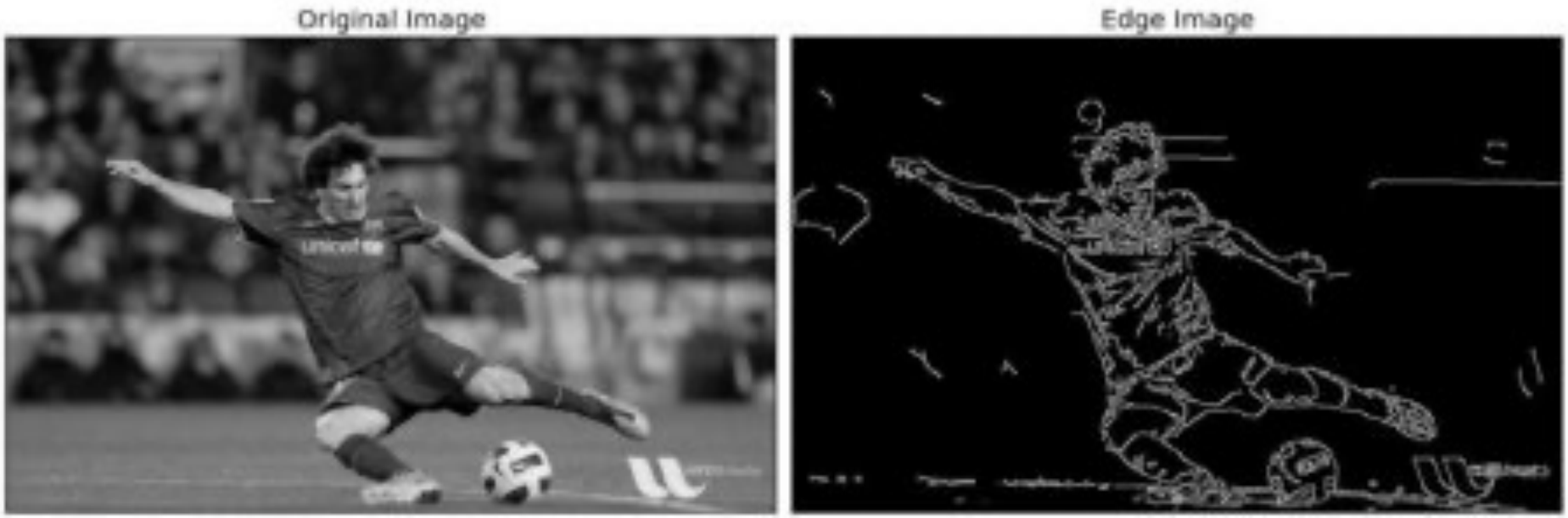
Canny Edge Detection

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('messi5.jpg',0)
edges = cv2.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()
```

AncoraSIR.com

OpenCV

Canny Edge Detection



AncoraSIR.com

OpenCV

Contours in OpenCV

- Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity.
- Useful tool for shape analysis and object detection and recognition.
  - For better accuracy, use binary images. So before finding contours, apply threshold or canny edge detection.

```
im = cv2.imread('test.jpg')
imgray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
ret,thresh = cv2.threshold(imgray,127,255,0)
image, contours, hierarchy =
cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
```

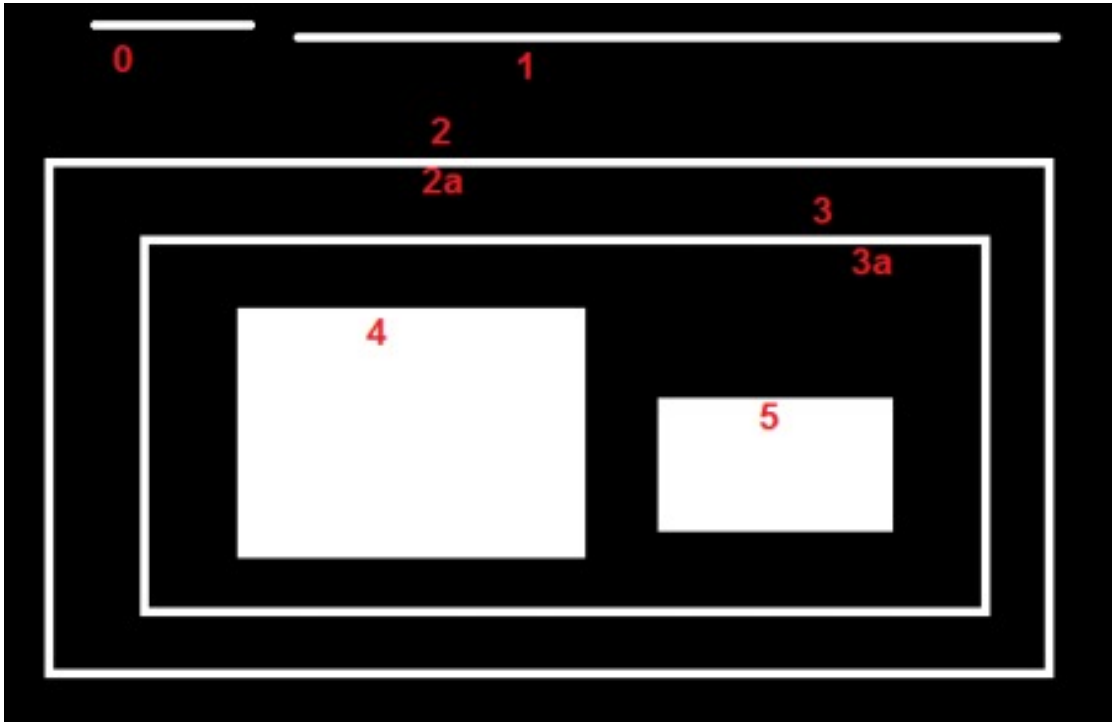
AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

Contours in OpenCV

- Contour Hierarchy: how one contour is connected to each other, outer one as parent and inner one as child
- Contour Retrieval Mode: RETR_LIST, RETR_TREE



AncoraSIR.com

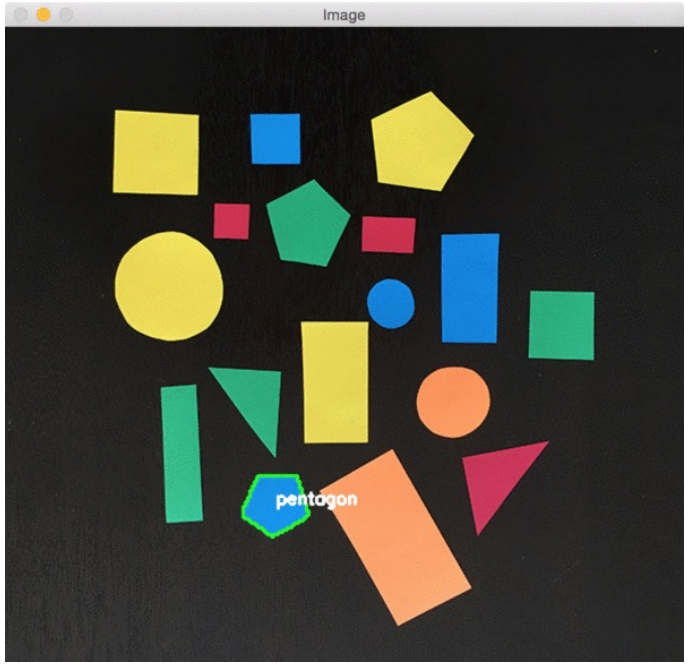
SUSTech
Southern University of Science and Technology

OpenCV

Contour Approximation for Shape Detection

- Contour approximation is predicated on the assumption that a curve can be approximated by a series of short line segments.
- Contour approximation is implemented in OpenCV via thecv2.approxPolyDP

```
# if the shape is a triangle, it will have 3 vertices
if len(approx) == 3:
    shape = "triangle"

# if the shape has 4 vertices, it is either a square or
# a rectangle
elif len(approx) == 4:
    # compute the bounding box of the contour and use the
    # bounding box to compute the aspect ratio
    (x, y, w, h) = cv2.boundingRect(approx)
    ar = w / float(h)

    # a square will have an aspect ratio that is approximately
    # equal to one, otherwise, the shape is a rectangle
    shape = "square" if ar >= 0.95 and ar <= 1.05 else "rectangle"

# if the shape is a pentagon, it will have 5 vertices
elif len(approx) == 5:
    shape = "pentagon"

# otherwise, we assume the shape is a circle
else:
    shape = "circle"

# return the name of the shape
return shape
```



AncoraSIR.com

OpenCV

Template Matching

- Template Matching is a method for searching and finding the location of a template image in a larger image.
- It simply slides the template image over the input image (as in 2D convolution) and compares the template and patch of input image under the template image.
- Several comparison methods are implemented in OpenCV

AncoraSIR.com

SUSTech
Southern University of Science and Technology

OpenCV

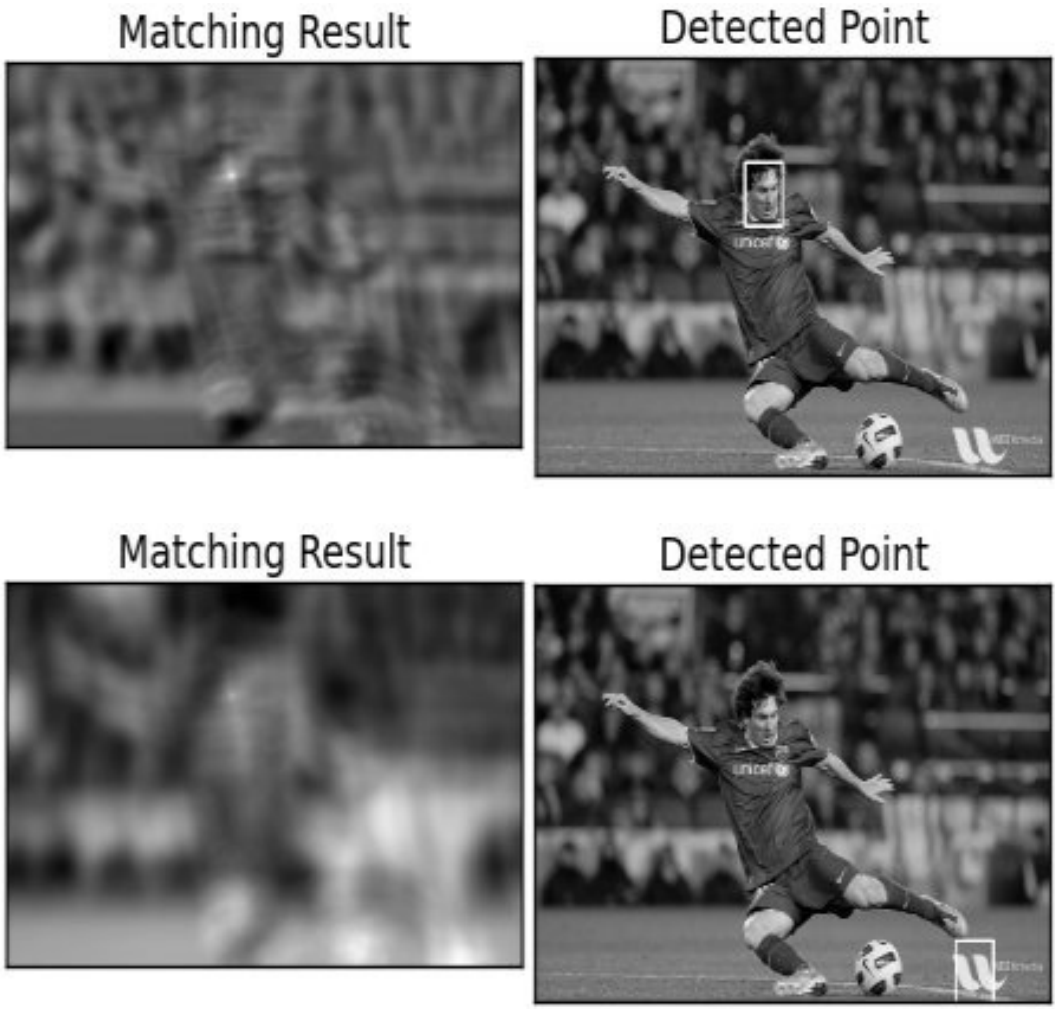
Template Matching

```
img = cv2.imread('messi5.jpg',0)
img2 = img.copy()
template = cv2.imread('template.jpg',0)
w, h = template.shape[::-1]

# All the 6 methods for comparison in a list
methods = ['cv2.TM_CCOEFF', 'cv2.TM_CCOEFF_NORMED', 'cv2.TM_CCORR',
            'cv2.TM_CCORR_NORMED', 'cv2.TM_SQDIFF', 'cv2.TM_SQDIFF_NORMED']

for meth in methods:
    img = img2.copy()
    method = eval(meth)

    # Apply template Matching
    res = cv2.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(res)
```

AncoraSIR.com

OpenCV

Template Matching



AncoraSIR.com

OpenCV

Hough Circle Transform

- Hough Transform to find circles in an image

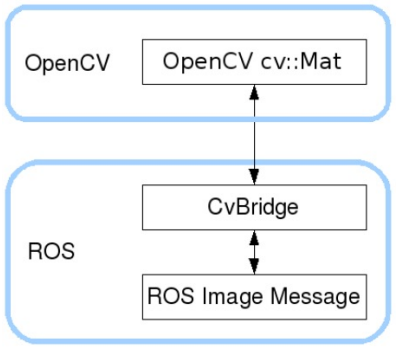

```
img = cv2.imread('opencv_logo.png',0)
img = cv2.medianBlur(img,5)
cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)

circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,20,
                           param1=50,param2=30,minRadius=0,maxRadius=0)

circles = np.uint16(np.around(circles))
for i in circles[0,:]:
    # draw the outer circle
    cv2.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2)
    # draw the center of the circle
    cv2.circle(cimg,(i[0],i[1]),2,(0,0,255),3)

cv2.imshow('detected circles',cimg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

AncoraSIR.com

OpenCV

ROS – OpenCV interface

- The OpenCV library is interfaced to ROS via ROS stack called vision_opencv consisting of
  - cv_bridge: converting between the OpenCV image data type and the ROS image message.
  - image_geometry: correct the geometry of the image using calibration parameters
- Image processing using ROS and OpenCV



- Subscribe the images from the camera driver from the topic `/usb_cam/image_raw` (`sensor_msgs/Image`)
- Convert the ROS images to OpenCV image type using `CvBridge`
- Process the OpenCV image using its APIs and find the edges on the image
- Convert the OpenCV image type of edge detection to ROS image messages and publish into the topic `/edge_detector/processed_image`

AncoraSIR.com

SUSTech
Southern University of Science and Technology

PCL

What is Point Cloud?

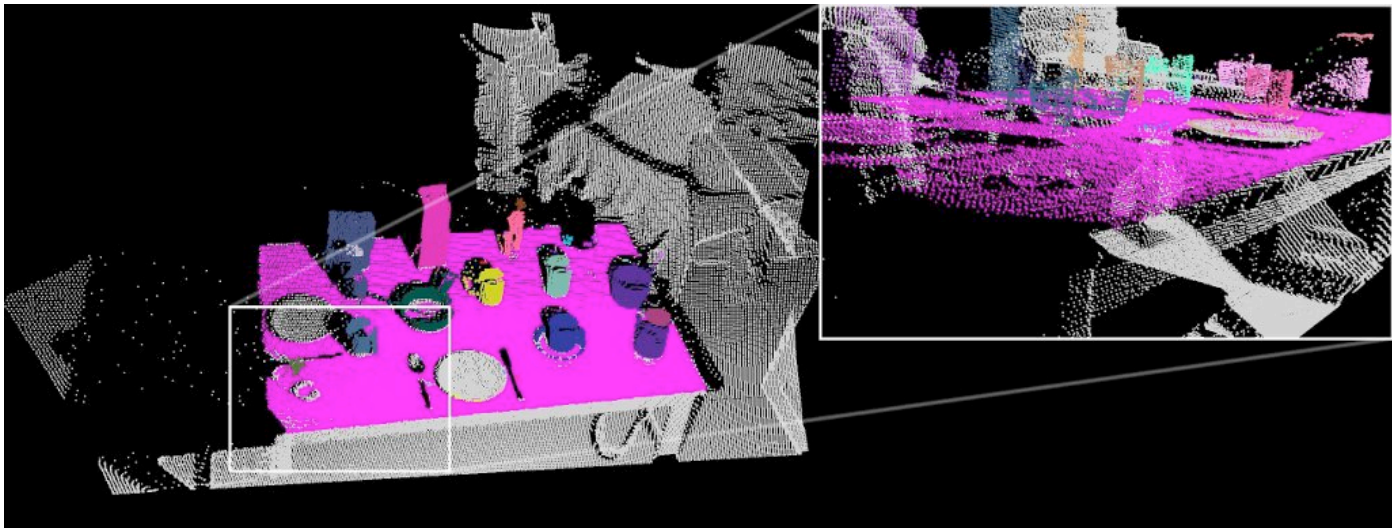
- Point Cloud = a collection of n-dimensional points, usually n = 3 (x, y, z) or 6 (x, y, z, r, g, b)

AncoraSIR.com

PCL

Point Cloud Usage

- Navigation / Obstacle avoidance
- Object recognition
- Grasping

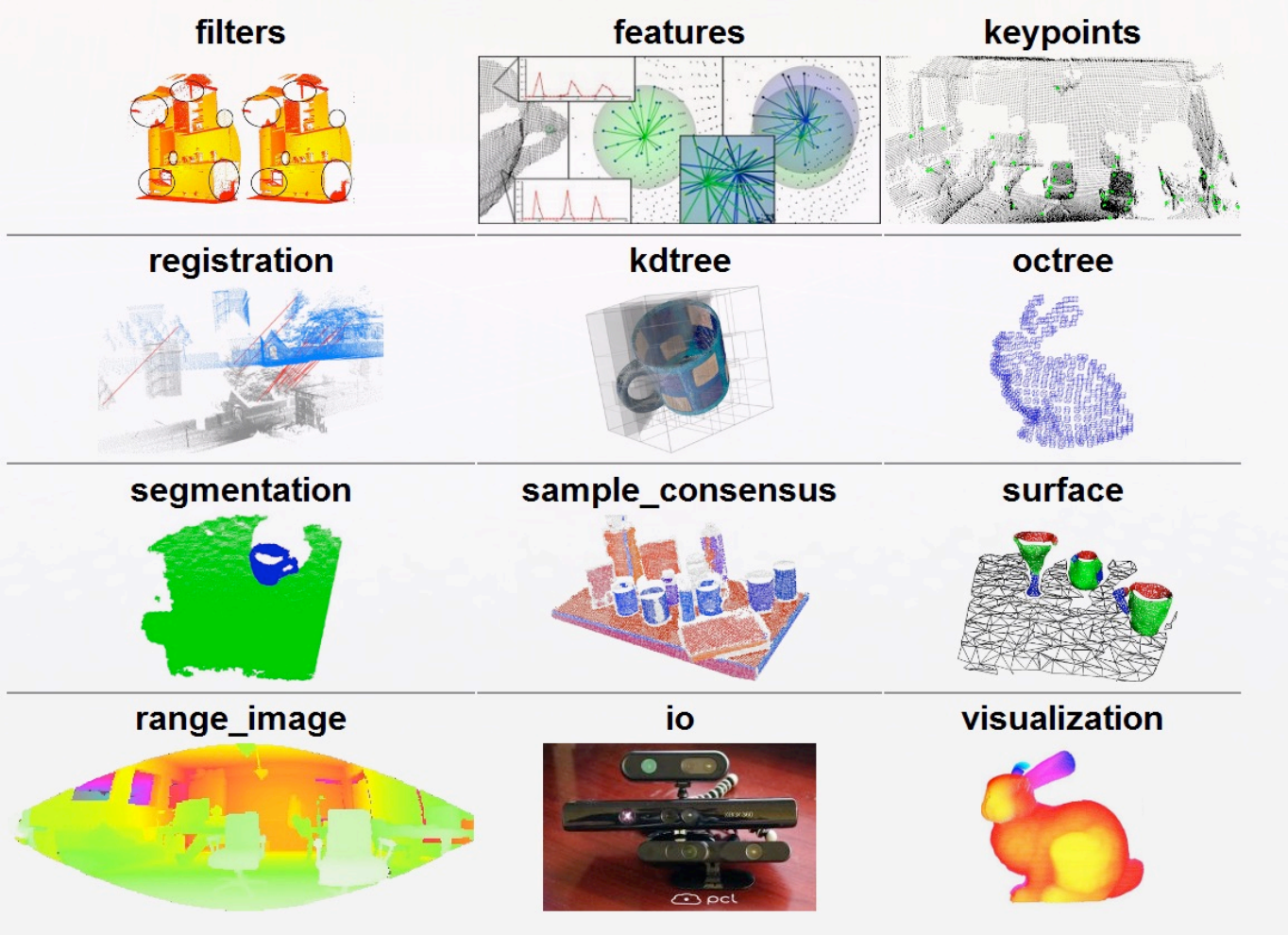


AncoraSIR.com

SUSTech
Southern University of Science and Technology

PCL

Point Cloud Library Modules



AncoraSIR.com

PCL

Point Cloud Library Modules

- pcl_filters library contains outlier and noise removal mechanisms for 3D point cloud data filtering applications.
- pcl_features library contains data structures and mechanisms for 3D feature estimation from point cloud data.
- pcl_registration library implements a plethora of point cloud registration algorithms for both organized and unorganized (general purpose) datasets.
- pcl_segmentation library contains algorithms for segmenting a point cloud into distinct clusters.

AncoraSIR.com

http://pointclouds.org/documentation/

SUSTech
Southern University of Science and Technology

Homework

- Write a ros node of image processing: locate the pick items from RGB images and compute the pick pose reference to the camera coordinate.
- Prepare for the Project2:
  - Project2: Picking Robot with Vision
  - Codes and instructions can be found at
  - https://github.com/ancorasir/BionicDL-CobotLearning-Project2

AncoraSIR.com

SUSTech
Southern University of Science and Technology