Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technische Universität München

RoboCup@Home Practical Course

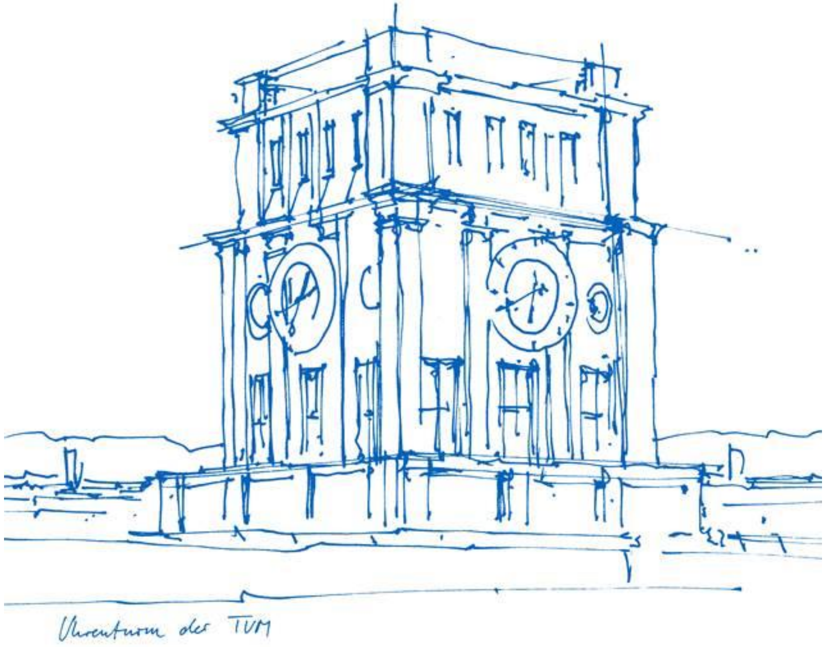
Dr. Pablo Lanillos

Technical University of Munich

Department of Electrical and Computer Engineering

Chair for Cognitve Systems

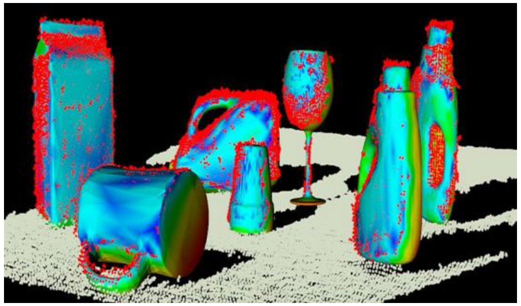
Munich, 22. November 2017



RoboCup@Home Practical Course

Tutorial: Image processing

Dr. Karinne Ramirez-Amaro
Dr. Emmanuel Dean
Dr. Pablo Lanillos
M.Sc. Roger Guadarrama
Dr. Gordon Cheng



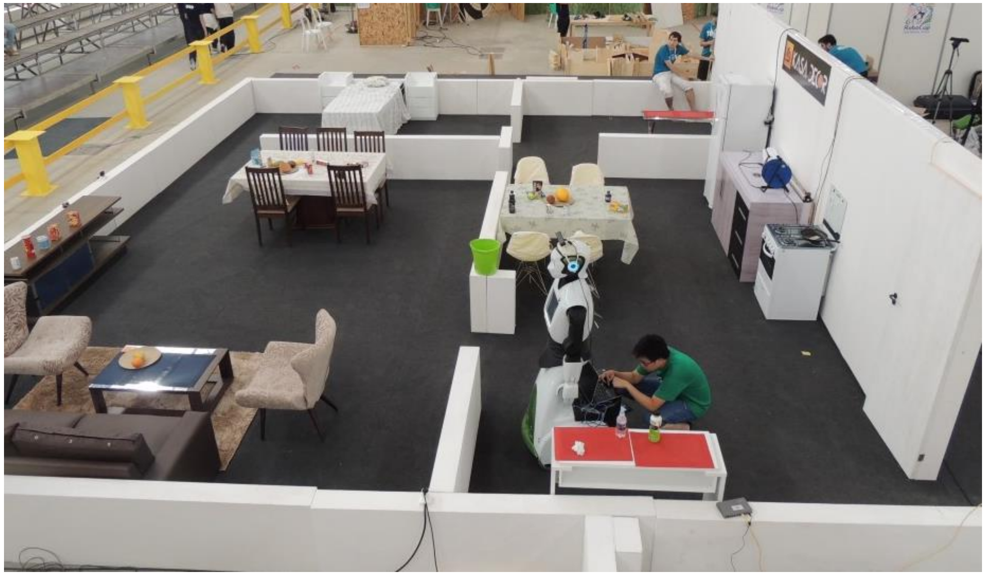
p.lanillos@tum.de
www.therobotdecision.com

Exercises

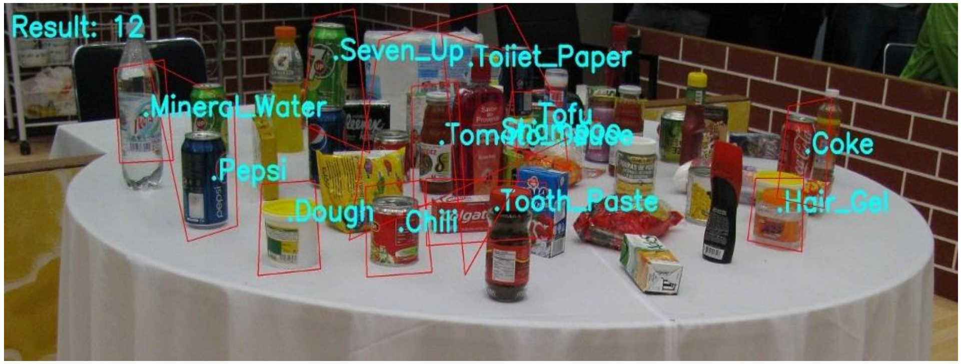
Email: robocup.atHome.ics@gmail.com

- Compress all the folder containing the C++ nodes folders into one zip/rar/tar/gz file and name it as: Name_LastName_RCH_tutorial4.

RoboCup@Home scenarios



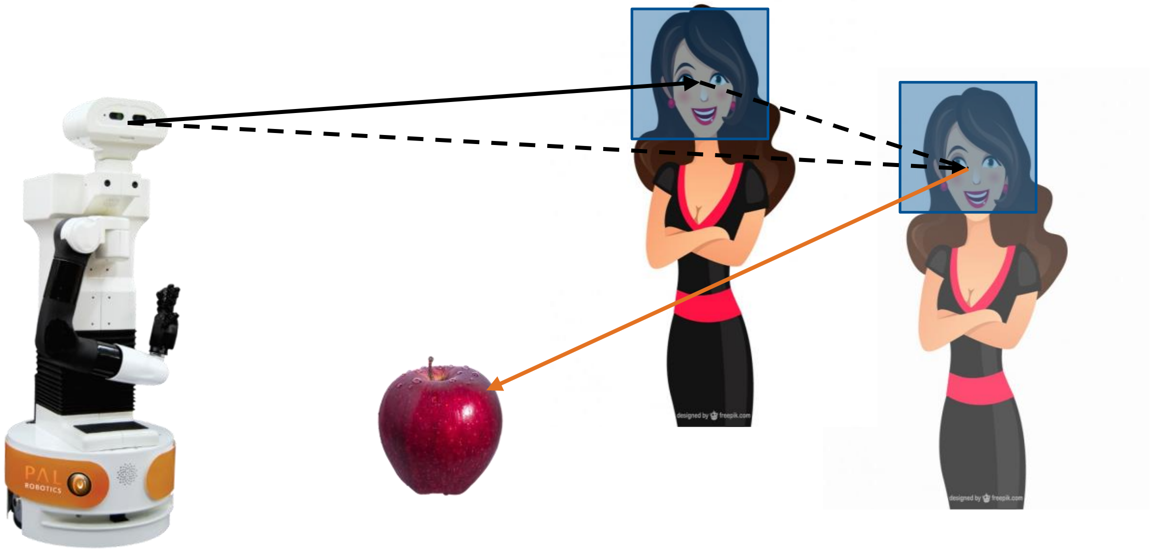
Robocup@home 2015



Homer Team Koblenz Uni.

Goal

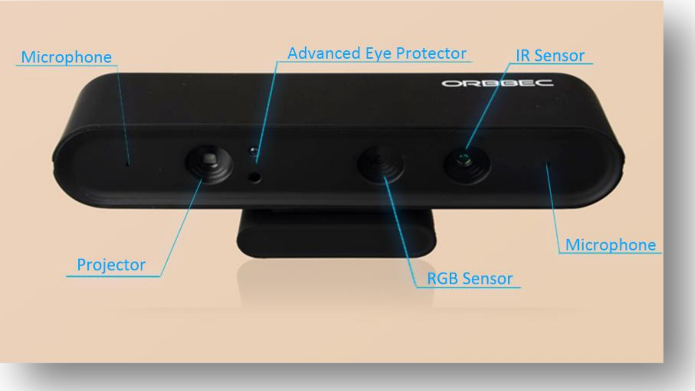
Human-robot interaction using vision



Implement computer vision algorithms with openCV in C++ and Python under ROS

Before starting: RGBD sensors

PrimeSense



https://orbbec3d.com/product-astra/

Hold on! What is an image?

Lanillos, P., Ferreira, J. F., Dias, J. (2015): Designing an Artificial Attention System for Social Robots. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.

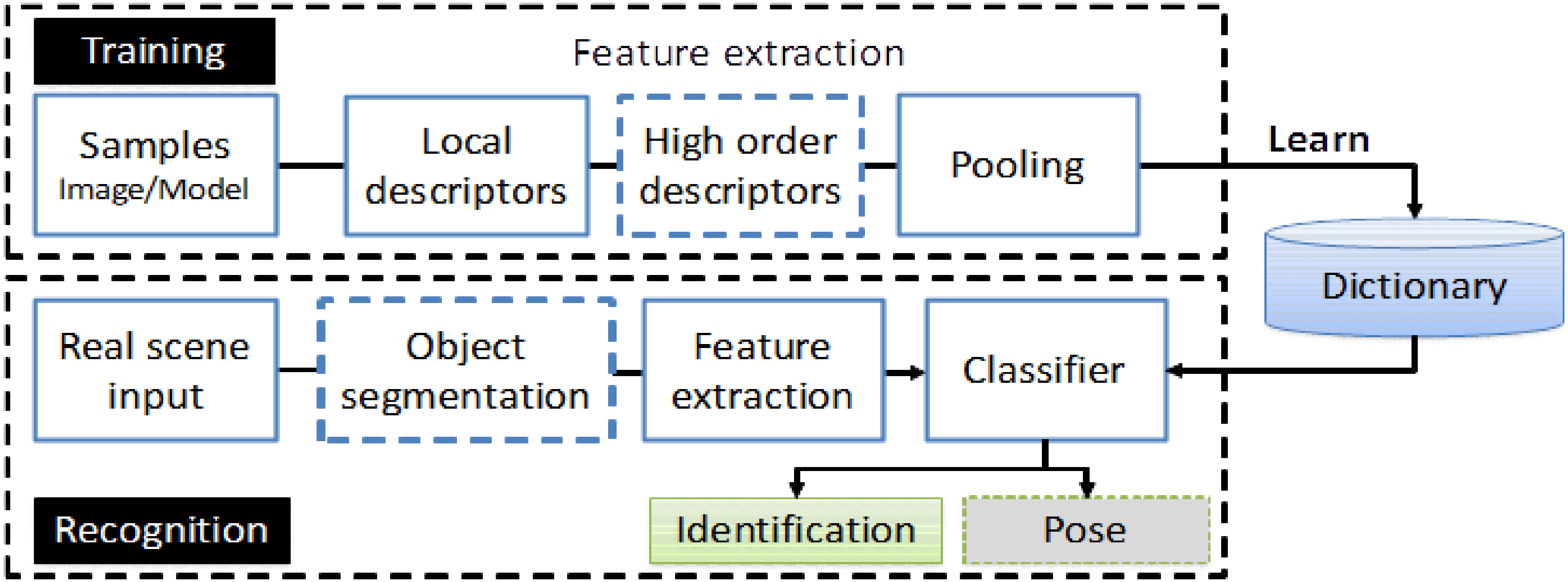
Excercises

1. Implement a C++ ROS node using the OpenCV face detection algorithm based on the code in python for **multiple** faces.
   face_detection.cpp

2. A) Implement a C++ ROS node that segments the image given a hue value.
   color_segementation.cpp
   B) Implement a C++ ROS node that segments the image into edges
   edges_segementation.cpp

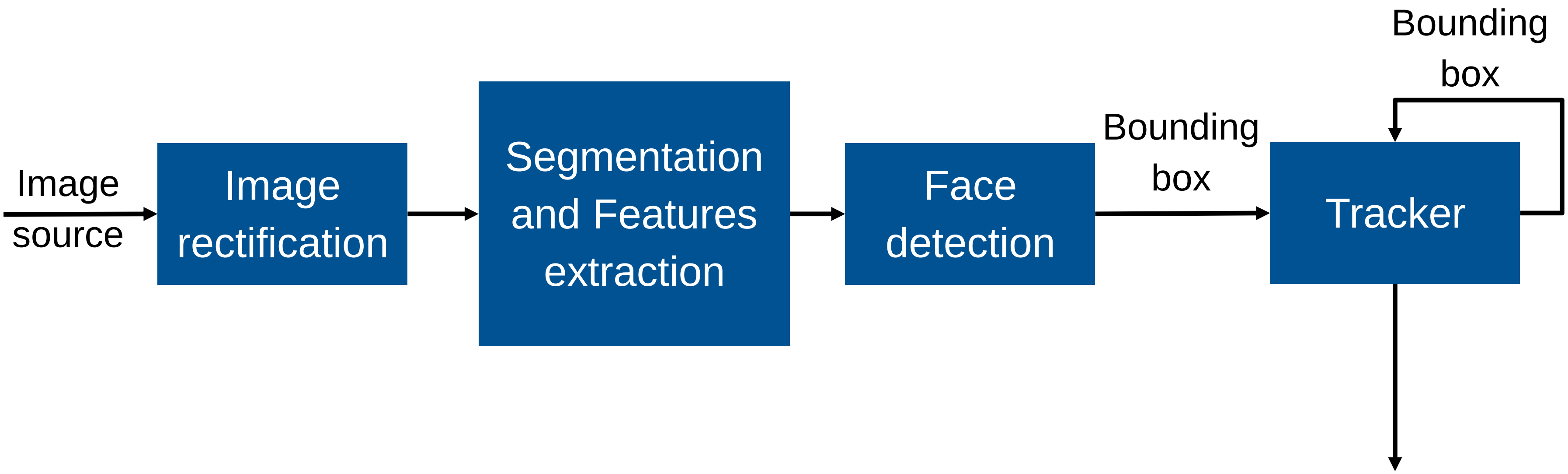
3. Integrate the tracking node provided with the face detector and colour segmentation.
   tracker.cpp

Classic pipeline



Adapted from: Wang, W., Chen, L., Liu, Z., Kühnlenz, K., & Burschka, D. (2015). Textured/textureless object recognition and pose estimation using RGB-D image. Journal of Real-Time Image Processing, 10(4), 667-682.

ROS pipeline for this tutorial



Exercise 1: Face detection

1. Go to face_detection_python directory
2. Complete the missing lines in the code at the FIXME keywords
3. Compile the code (place a CATKIN_IGNORE file in the other folders)

```
catkin_make
```

4. Run the kinect or asus or orbbec

```
roslaunch kinect2_bridge kinect2_bridge.launch
roslaunch openni2_launch openni2_launch.launch
```

5. Check the topics being published

```
rostopic list
```

6. Run the code

```
rosrun face_detection_python face_detection.py
```

7. Create folder inside the catkin directory and implement a ROS face detection C++ node

```
mkdir  <catkin_workspace>/src/face_detection
```

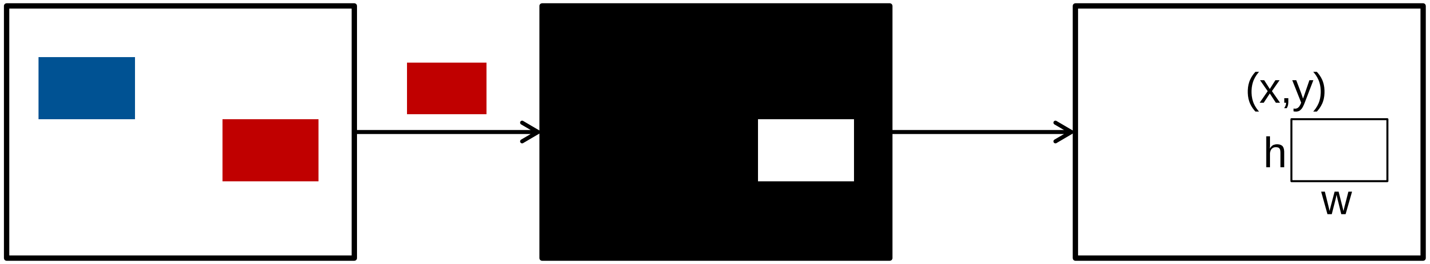
Input: Camera stream
Output: bounding box (Rect message)

Exercise 2: Color segmentation

1. Create folder inside the catkin directory

```
mkdir  <catkin_workspace>/src/segmentation
```

2. Implement hue color segmentation node: *color_segmentation.cpp*



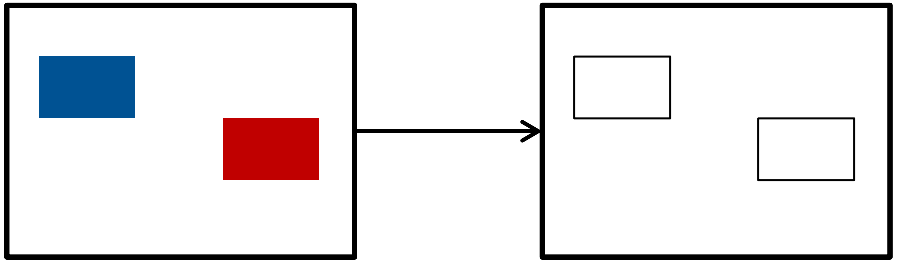
Input: Camera stream
Output: bounding boxes (Rect message)

Exercise 2: Edges segmentation

1. Create folder inside the catkin directory

```
mkdir  <catkin_workspace>/src/segmentation
```

2. Implement edge segmentation node: *edge_segmentation.cpp*



Input: Camera stream
Output: Image with edges

Exercise 3: Tracking integration

1. Go inside the /src/tracker
2. Check the ROS tracker C++ node in trackernode.cpp
3. Compile the code (remove the CATKIN_IGNORE file)

```
catkin_make
```

4. Run the kinect

```
roslaunch kinect2_bridge kinect2_bridge.launch
roslaunch openni2_launch openni2_launch.launch
```

5. Run the code

```
rosrun tracker trackernode
```

6. Connect both nodes through the topics

   face_detection/bb → tracker

   color_segmentation/bb → tracker

7. Create a launch file to run the detectors and the tracker at the same time:

   image_processing.lauch

Tip: Recording data in a rosbag

```
> roslaunch kinect2_bridge kinect2_bridge.launch (for kinect)
> roslaunch openni2_launch openni2_launch.launch (for asus or orbbec)
```

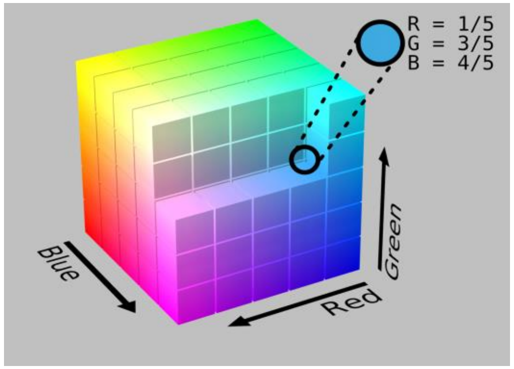
```
> rosrun rviz rviz → add image → image_rectified
```

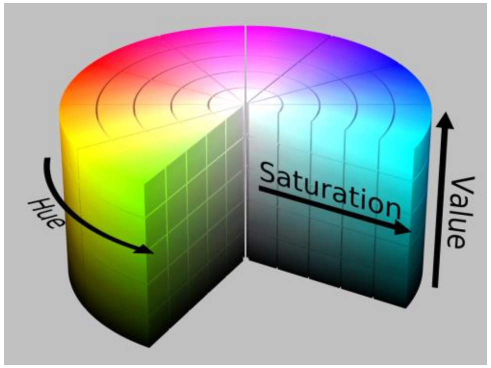
```
> rosbag record -a [-O session_name.bag]
```

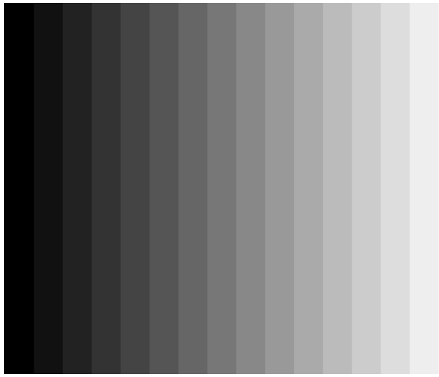
To play

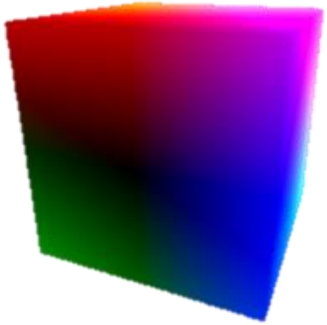
```
> rosbag play -l name_of_the_file
          (-l makes the bag to play in a loop)
```

Color



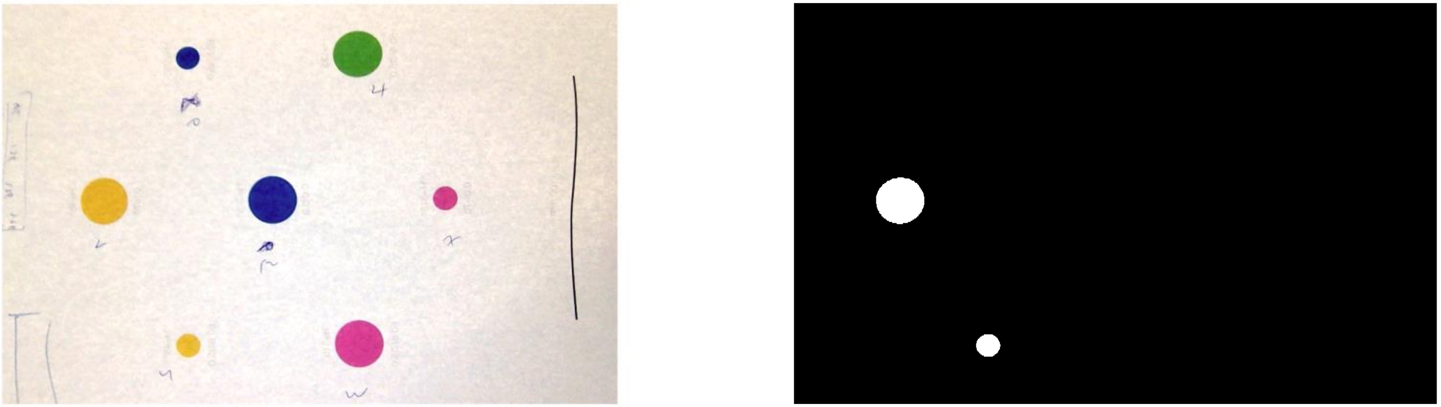






The YUV is represented by intensity (Y) and the color information: blue-green correlation (U) and red-green correlation (V).

Color segementation

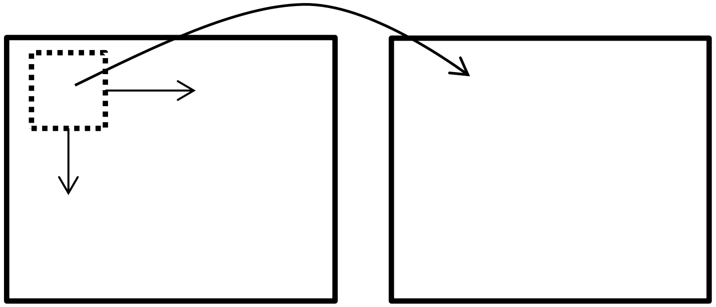


$$if\ image[i][j][R] \geq r_1\ \&\&\ image[i][j][R] \leq r_2\ \rightarrow$$

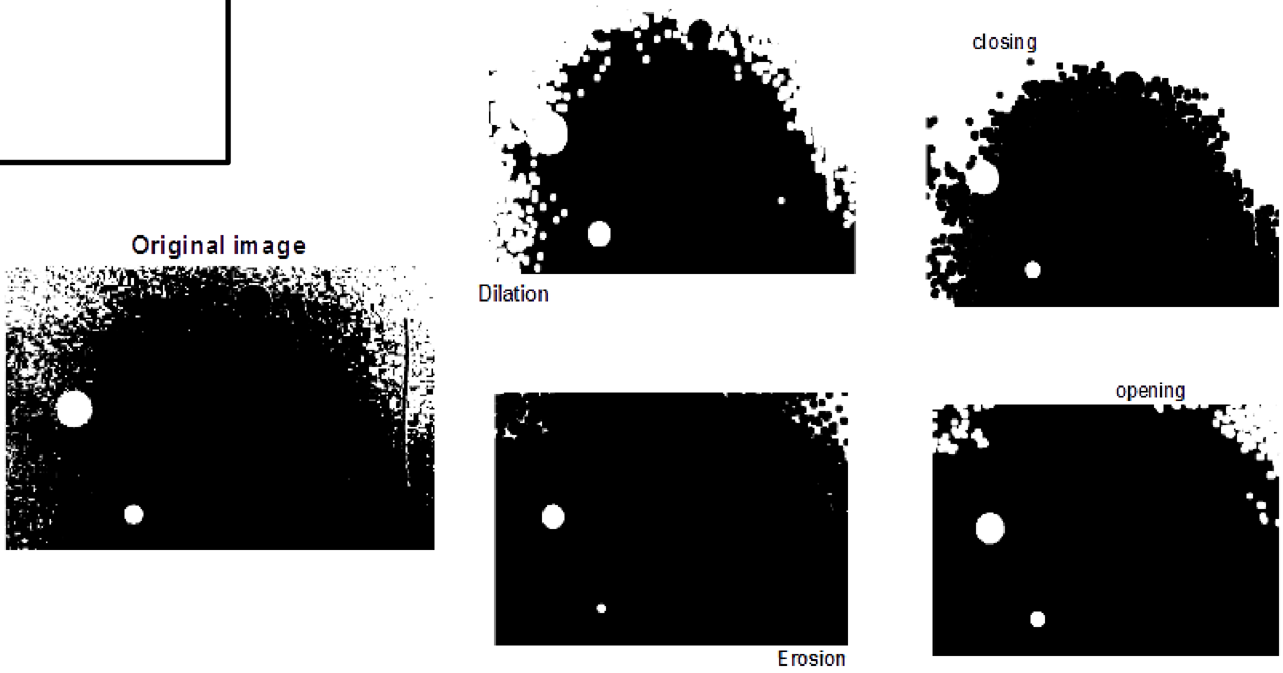
How we can program a color pixel classification with constant complexity O(1)?

Kernels and convolutions

$$I \circ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = ?$$

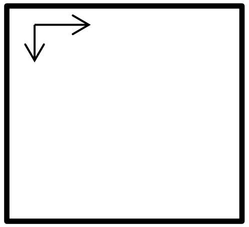


Kernel for dilation and erosion?



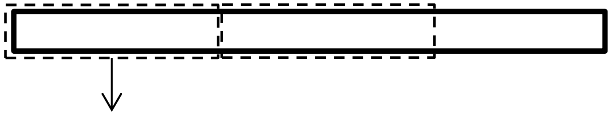
Traversing the image !

Hold on: What is an image?

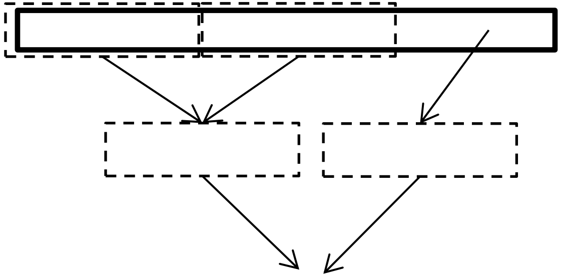


standard

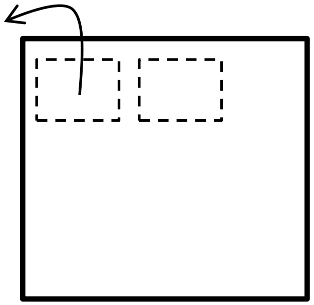
$idx = i * row + j$ Iterators and const. iterators

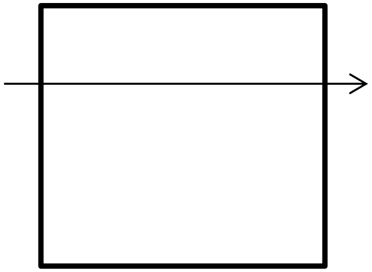


pyramidal processing

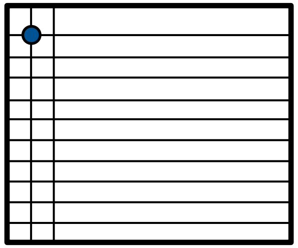


parallel operators





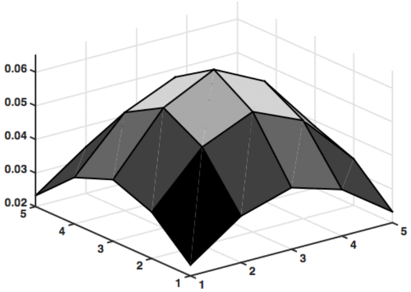
Line tracing



Subsampling

Kernels and Descriptors

Lena. Alexander Sawchukat



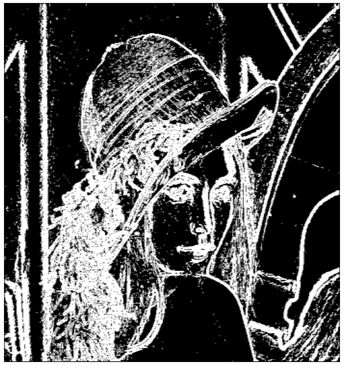
Gaussian filter

$$\begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$

Why gradient is important?

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

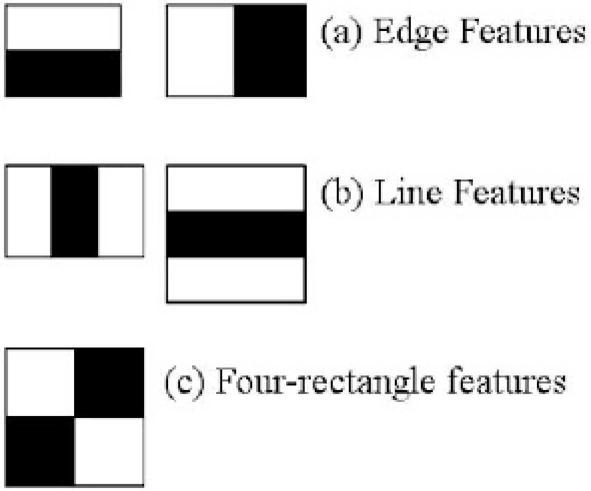
Horizontal Sobel filter



How is the vertical kernel filter?

Face descriptors

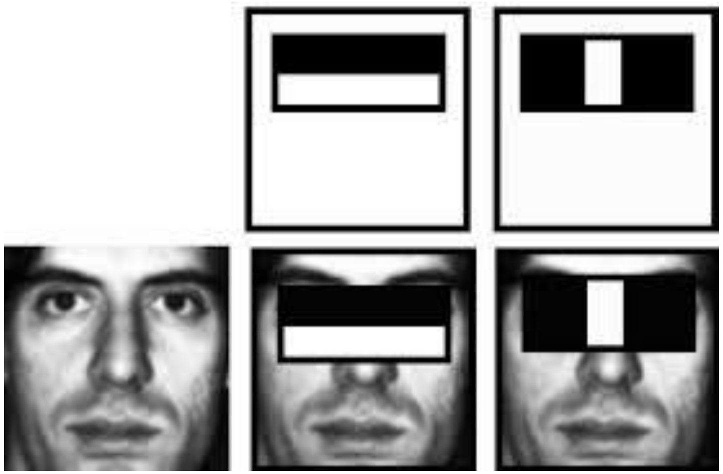
HAAR cascade classifier



(a) Edge Features

(b) Line Features

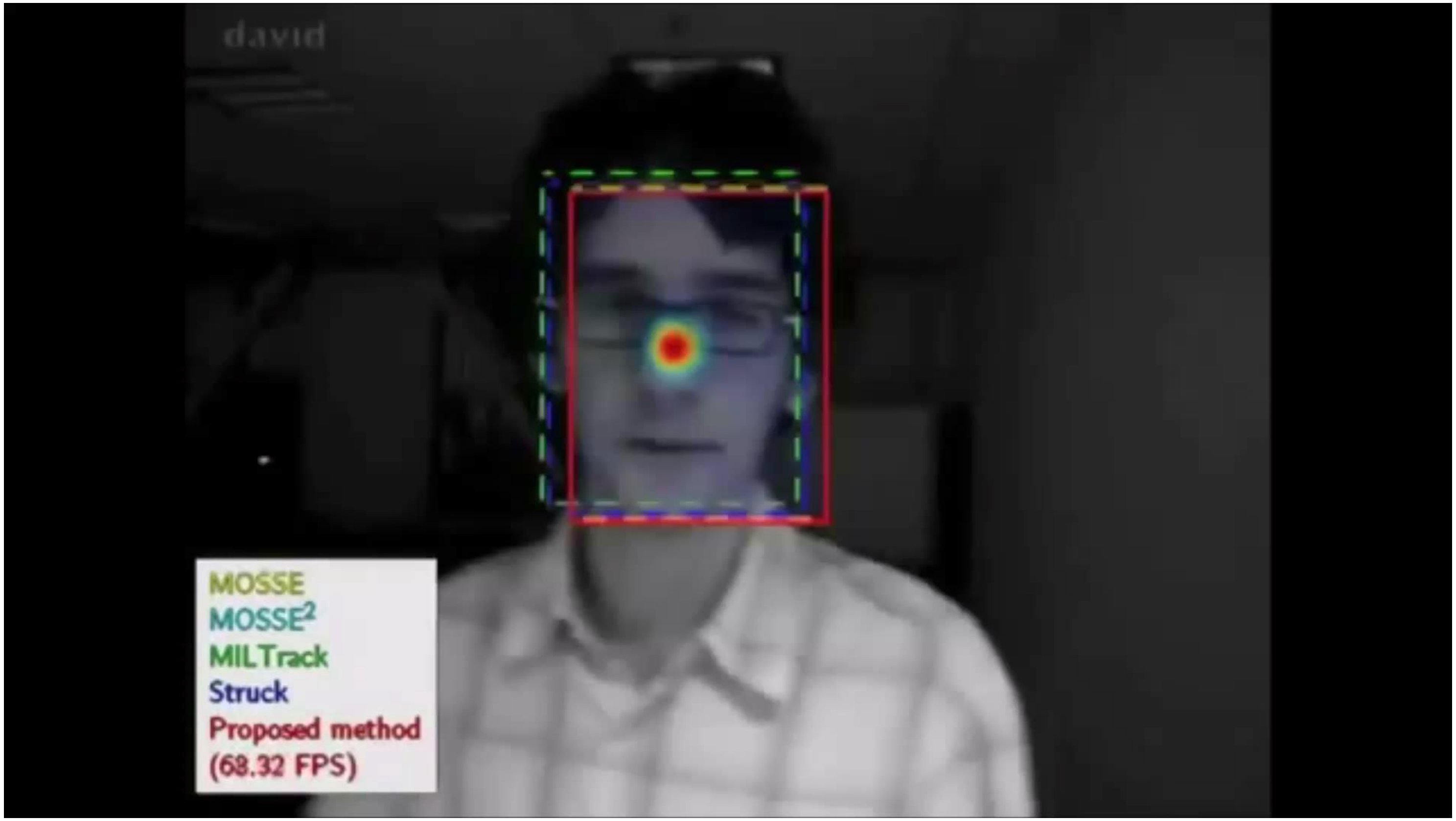
(c) Four-rectangle features



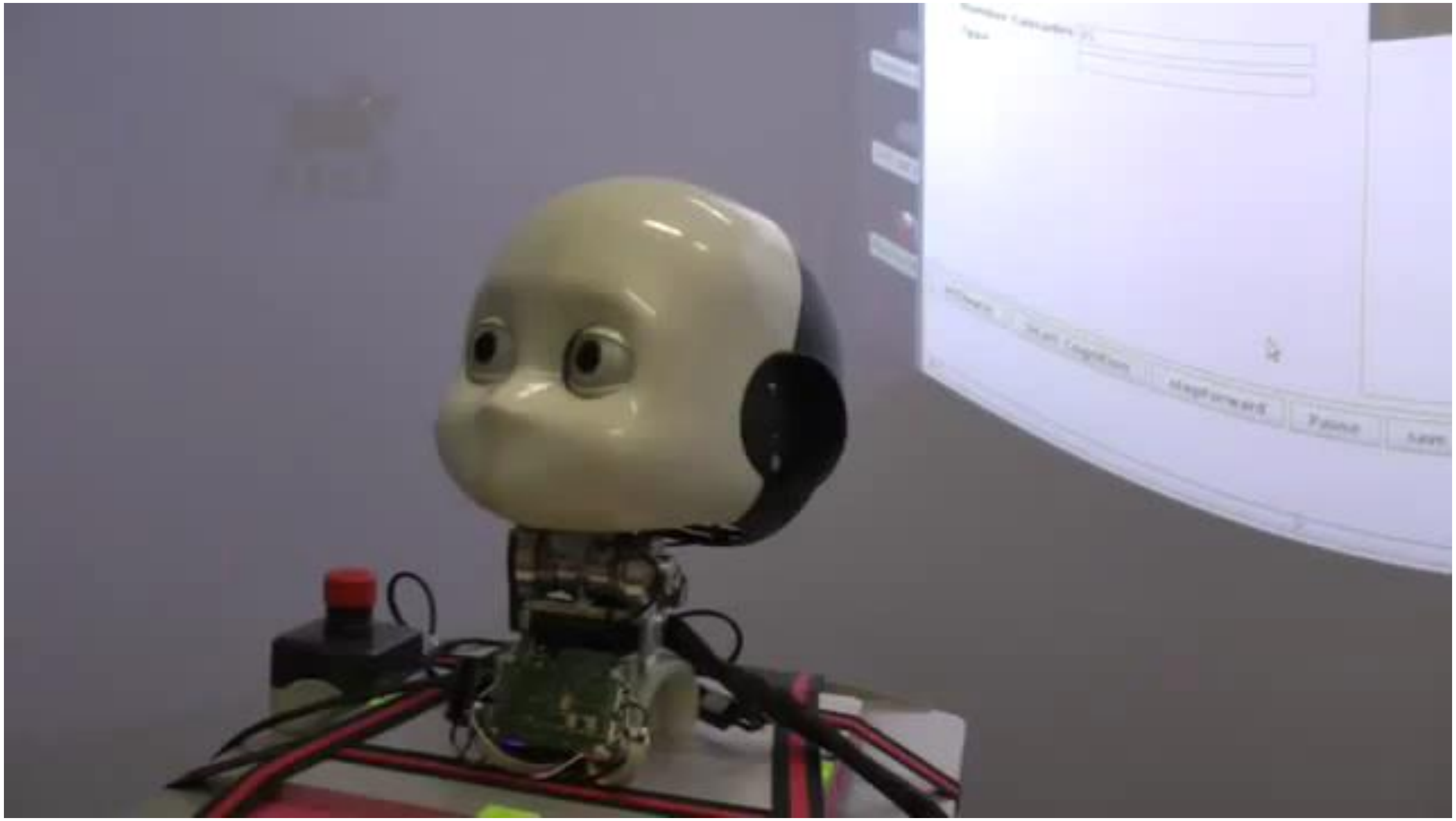
Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511).

Extra material Viola & Jones algorithm video explanation
https://www.youtube.com/watch?v=WfdYYNamHZ8

Tracking


david
MOSSE
MOSSE²
MILTrack
Struck
Proposed method
(68.32 FPS)

Simple human-robot interaction: Engage!



MIGHT THE PIXELS BE WITH YOU!