Human-Computer Interaction

Visual Perception of Humans

Luca Iocchi

DIAG, Sapienza University of Rome, Italy

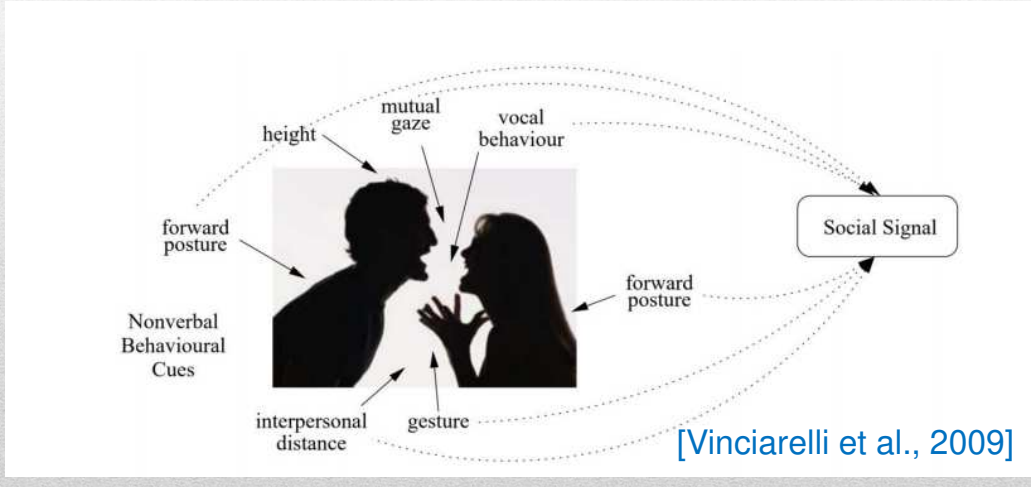
With contributions from D. D. Bloisi and A. Youssef

Context

Any computing device (machine) equipped with a camera can improve performance in human-machine interaction by "understanding" humans around it.

Social signals

Human-Machine Interaction ⇔ understanding and producing social signals



[Vinciarelli et al., 2009]

Applications

- **Personalized advertisement**
  - Advertisement screens in public spaces

Applications

- **Personalized user interfaces**
  - Biometric check points
  - Advanced UI

Applications

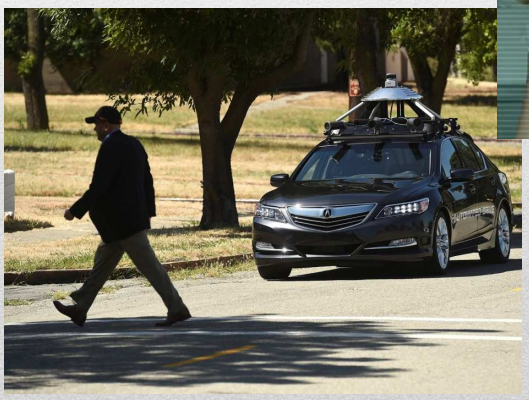
- **Augmented reality**
  - Virtual mirror
  - Virtual 3D environments

Applications

- **Entertainment**
  - Video games
  - Digital media

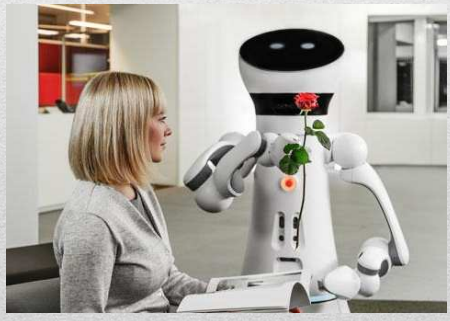
Applications

- **Safety & security**
  - Video-surveillance
  - Autonomous vehicles



Applications

- **Human-robot interaction**
  - Social robots
  - Service robots
  - Co-workers



Readings

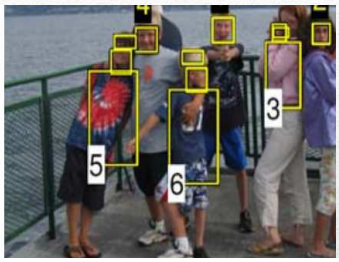
[Vinciarelli et al., 2009] Alessandro Vinciarelli, Maja Pantic, Hervé Bourlard. Social signal processing: Survey of an emerging domain. Image and Vision Computing, Volume 27, Issue 12, 2009. https://doi.org/10.1016/j.imavis.2008.11.007

Visual Perception for HRI/HCI

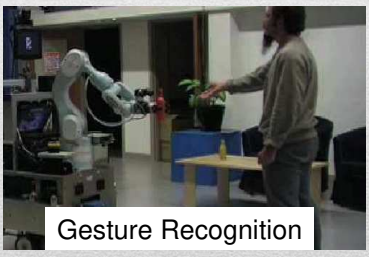
- Person detection, recognition, identification, and tracking
- Face detection, recognition
- Activity recognition
- Gestures
- Expressions
- Crowd analysis

Computer Vision for HRI/HCI

Person tracking



Face detection/recognition



Gesture Recognition

Virtual Reality

Some images from Computer Vision: Algorithms and Applications

Flow of processing

input data

OBJECT DETECTION

OBJECT RECOGNITION

time → T_i T_{i+1}

id

TRACKING

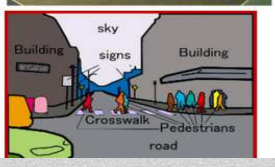
BEHAVIOR AND ACTIVITIES ANALYSIS

John is leaving the scene

Intelligent distributed surveillance systems: a review
M. Valera and S.A. Velastin

Frame Vision Problems

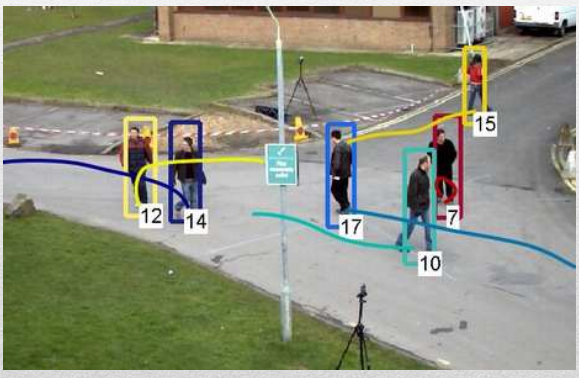
- classification
- localization
- detection
- segmentation



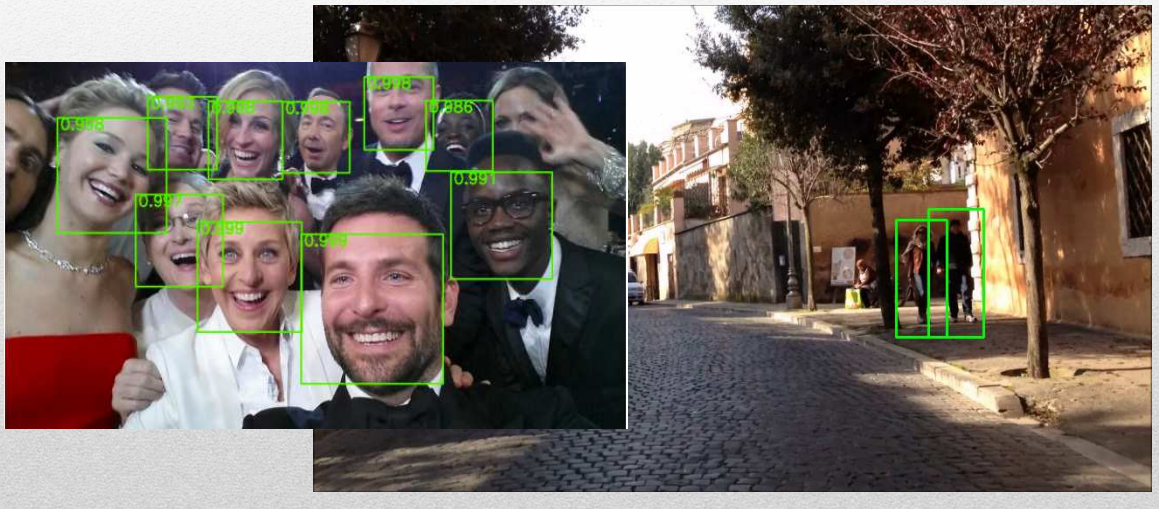
difficulty

Temporal Vision Problems

- Tracking (multi-person/object)
- Behavior recognition
- Scene understanding

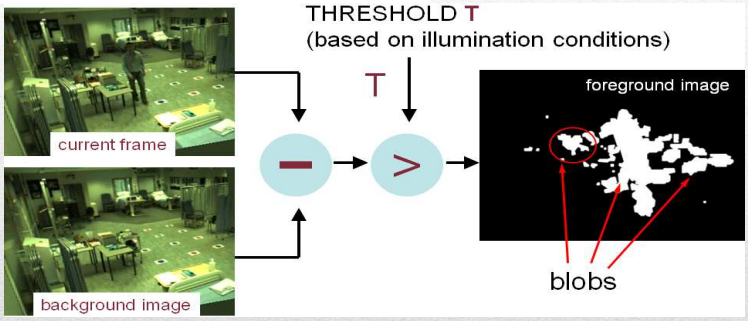


Face/Person Detection



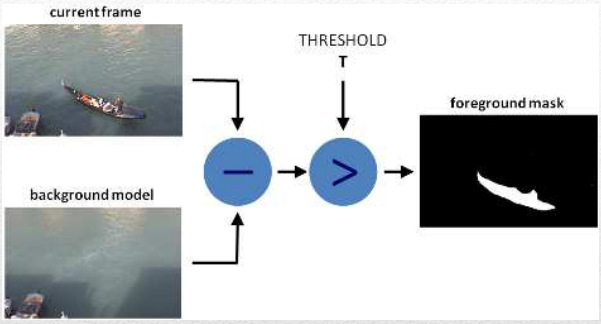
People Detection – two Approaches

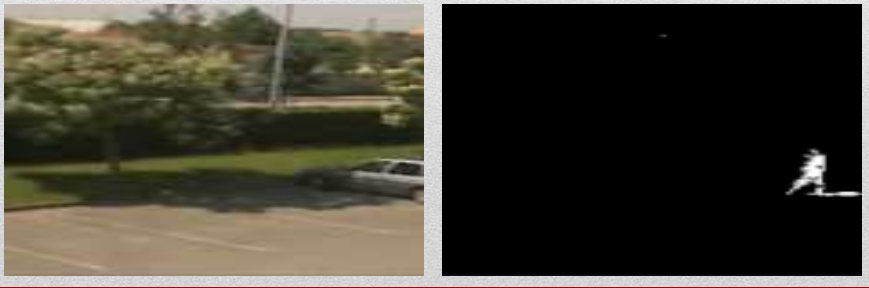
Background Subtraction



Classification-based detection

Background Subtraction





Gaussian Mixture Model

input video sequence

bg model

Each point in the bg model is a mixture pdf that is a linear combination of n Gaussian pdfs.

Evaluation

- Average ranking accross categories : (sum of ranks for all categories) / (number of categories)
- Average ranking : (rank:Recall + rank:Spec + rank:FPR + rank:FNR + rank:PWC + rank:FMeas
- TP : True Positive
- FP : False Positive
- FN : False Negative
- TN : True Negative
- Re (Recall) : TP / (TP + FN)
- Sp (Specificity) : TN / (TN + FP)
- FPR (False Positive Rate) : FP / (FP + TN)
- FNR (False Negative Rate) : FN / (TP + FN)
- PWC (Percentage of Wrong Classifications) : 100 * (FN + FP) / (TP + FN + FP + TN)
- F-Measure : (2 * Precision * Recall) / (Precision + Recall)
- Precision : TP / (TP + FP)
- FPR-S : Average False positive rate in hard shadow areas

Readings

C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999

Image Features

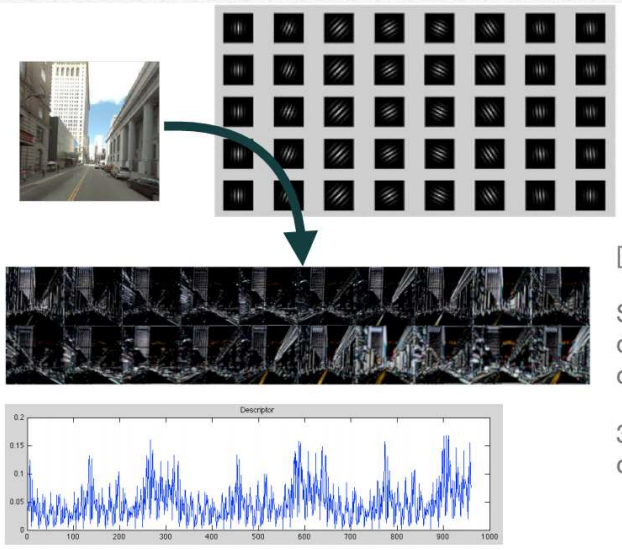
Global features

- GIST

Local features

- Corners (Harris, Shi-Tomasi, FAST, …)
- Scale invariant (SIFT, SURF, …)
- Binary (BRIEF, ORB, …)

GIST

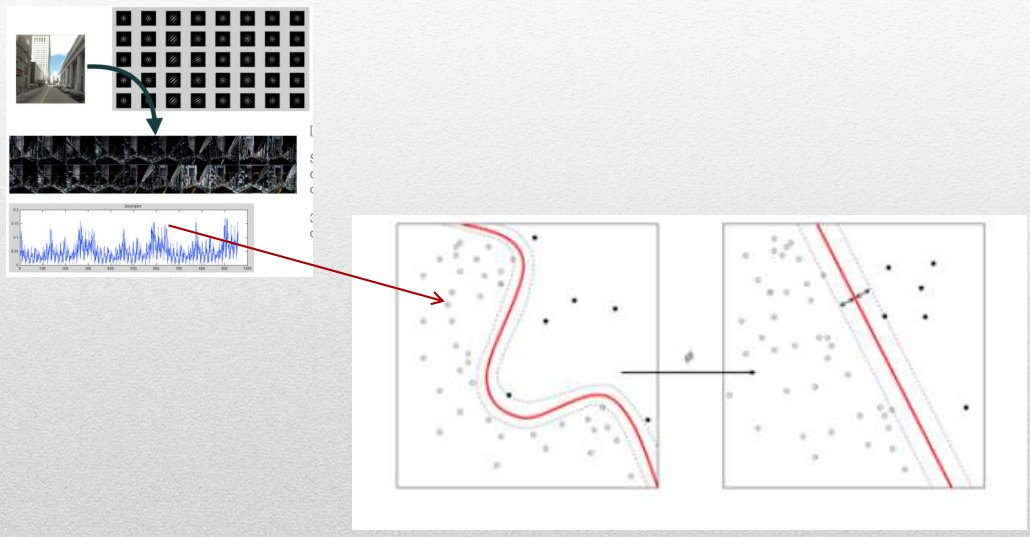


[Oliva&Torralba'01,06]

Several Gabor filters different scales & orientations

320 x 3 (RGB) descriptor

GIST + SVM



Histogram Of Oriented Gradients (HOG)

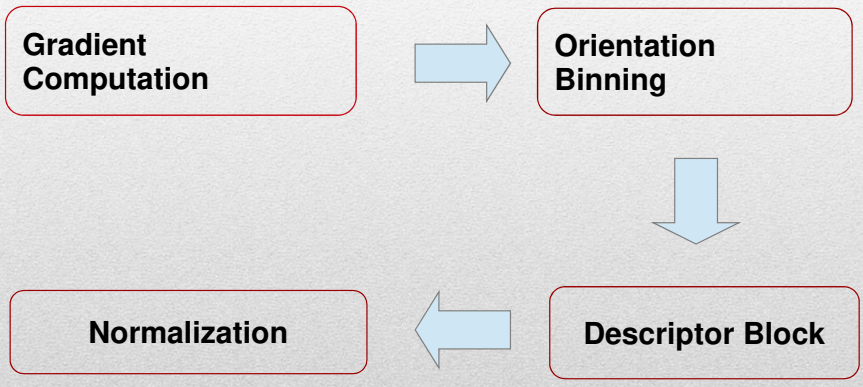
Sliding window technique for object detection in images

- Shape and appearance

Features Descriptor

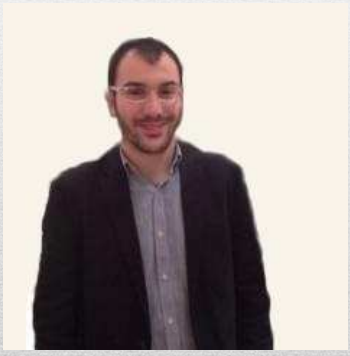
- Dense features extraction
- Local overlapping

HOG steps

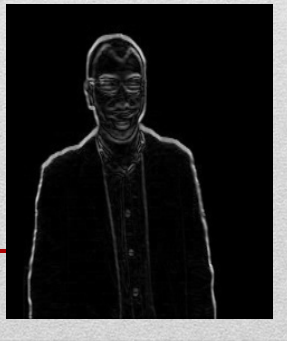


Gradient Computation

1-D mask centered operator
- X-filtering and Y-filtering



Gradient Vector
- magnitude and direction



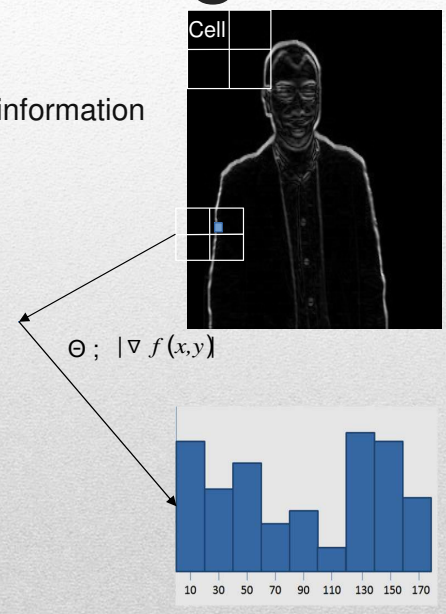
Orientation Binning

Cell creation

- **Cell** : group of pixels used to collect gradient information locally (e.g., 8x8 pixels for people detection)

Histograms channels

- Unsigned gradients
  - Separation over 0 to 180 degrees
  - 9 bins and 20 degrees for each bin
- Signed gradients
  - Separation over 0 to 360 degrees

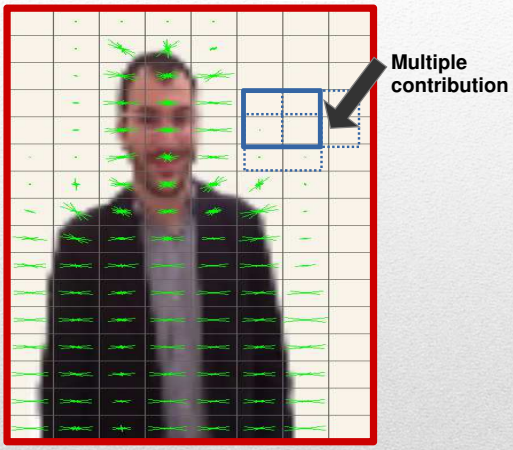


Descriptor Block

Normalized histogram for a group of cells located in the same block.

Same Cell contributes many times through the overlapping proceeds

- Block stride = 8x8 pixels.



Detection window : size of 64x128 pixels.

Block descriptor : size of 2 cells = 16 x 16 pixels.

Cell : size of 8x8 pixels.

People Detection Methods

		Features							Learning			Detection Details						Implementation			
		gradient hist.	gradients	grayscale	color	texture	self-similarity	motion	classifier	feature learn.	part based	non-maximum suppression	model height (in pixels)	scales per octave	frames per second (fps)	log-average miss rate	training data	original code	full image evaluation	publication
VJ	[44]			✓					AdaBoost			MS	96	~14	.447	95%	INRIA			'04
SHAPELET	[33]		✓	✓					AdaBoost	✓		MS	96	~14	.051	91%	INRIA			'07
POSEINV	[70]	✓							AdaBoost			MS	96	~18	.474	86%	INRIA	✓		'08
LATSVM-V1	[71]	✓							latent SVM		✓	PM	80	10	.392	80%	PASCAL	✓	✓	'08
FTRMINE	[67]	✓	✓	✓	✓				AdaBoost	✓		PM	100	4	.080	74%	INRIA	✓		'07
HIKSVM	[34]	✓							HIK SVM			MS	96	8	.185	73%	INRIA	✓		'08
HOG	[7]	✓							linear SVM			MS	96	~14	.239	68%	INRIA	✓		'05
MULTIFTR	[56]	✓		✓					AdaBoost			MS	96	~14	.072	68%	INRIA	✓	✓	'08
HOGLBP	[59]	✓				✓			linear SVM			MS	96	14	.062	68%	INRIA	✓	✓	'09
LATSVM-V2	[72]	✓							latent SVM		✓	PM	96	10	.629	63%	INRIA	✓	✓	'09
PLS	[69]	✓			✓	✓			PLS+QDA	✓		PM*	96	~10	.018	62%	INRIA	✓	✓	'09
MULTIFTR+CSS	[28]	✓					✓		linear SVM			MS	96	~14	.027	61%	TUD-MP	✓	✓	'10
FEATSYNTH	[68]	✓				✓			linear SVM	✓	✓	–	96	–	–	60%	INRIA	✓	✓	'10
FPDW	[63]	✓	✓	✓	✓				AdaBoost			PM*	100	10	6.492	57%	INRIA	✓	✓	'10
CHNFTRS	[29]	✓	✓	✓	✓				AdaBoost			PM*	100	10	1.183	56%	INRIA	✓	✓	'09
MULTIFTR+MOTION	[28]	✓					✓	✓	linear SVM			MS	96	~14	.020	51%	TUD-MP	✓	✓	'10

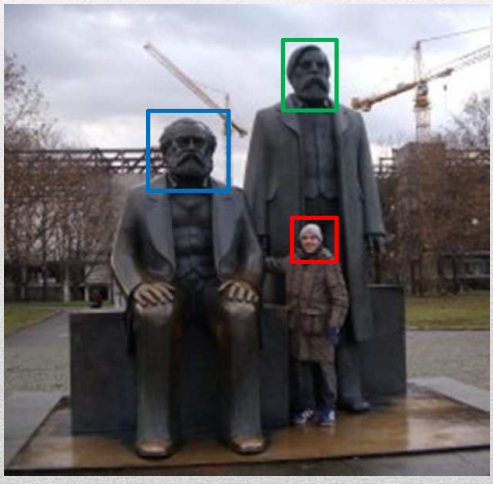
Accuracy vs real-time performance

Readings

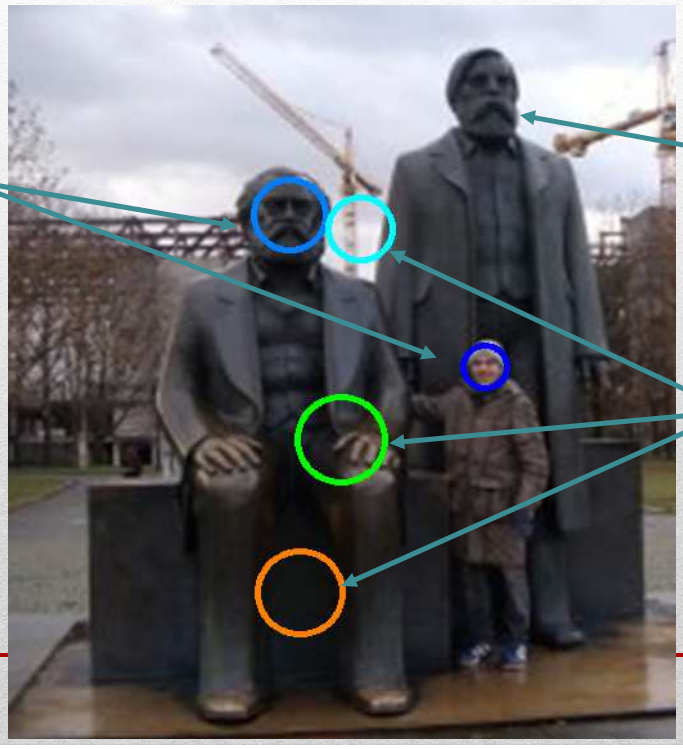
N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.886-893, vol. 1, 2005

Face Detection

Find regions in the image that contain instances of faces

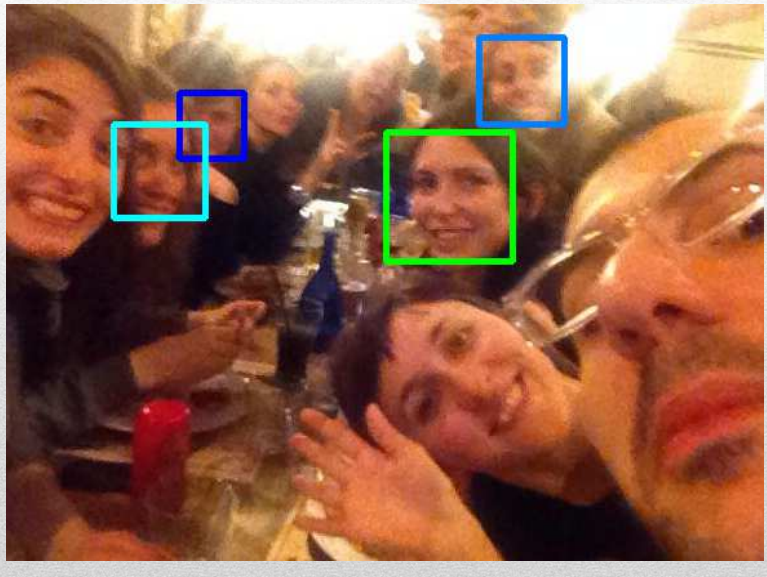


Detection Issues

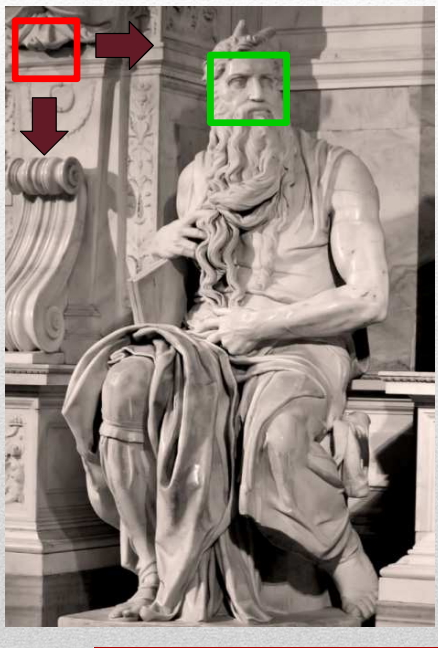


Additional Issues

- Rotation
- Blurring
- Illumination
- Occlusion
- Glasses
- …



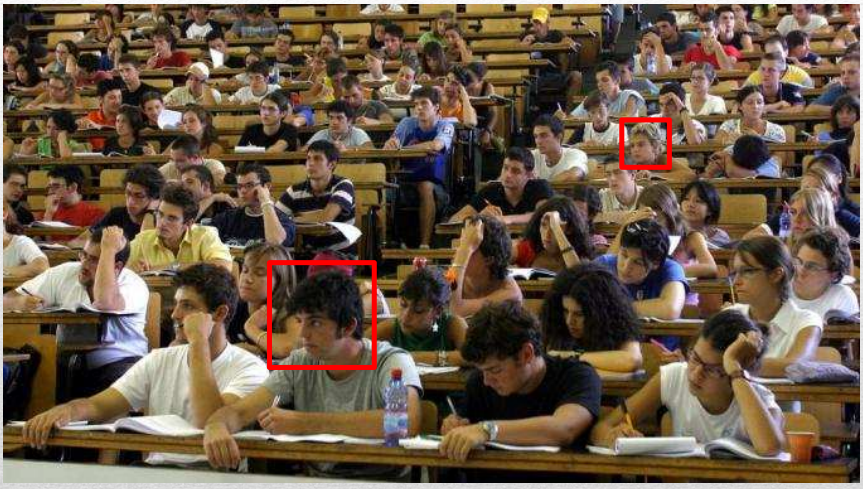
Sliding window search



Slide a window (e.g., 30x30) across the image and evaluate the current portion of the image w.r.t. the object model at every location

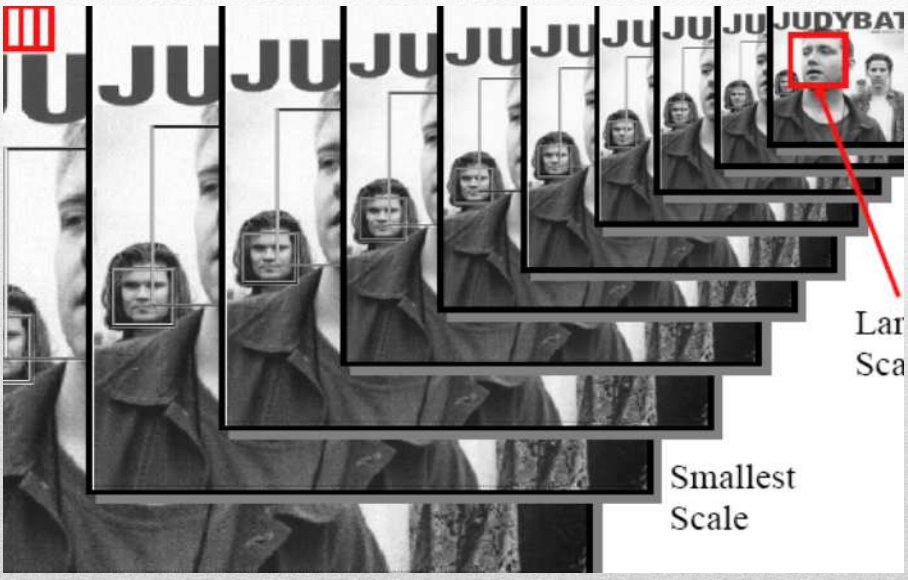
We assume that the number of locations where the object is present is (very) small

Multiple scale search



Multiple scale search

Image pyramid



Object Detection Steps

1. **Feature Computation** — Which features? How can they be computed as quickly as possible?
2. **Feature Selection** — What are the most discriminating features?
3. **Detection (in real time)** — Must focus on potentially positive areas

The Viola and Jones Method

- **Very popular method**
- **Recognition is very fast (e.g., real-time for digital cameras)**
- **Key contributions**
  1. **Integral image for fast feature extraction**
  2. **Boosting (Ada-Boost) for face detection**
  3. **Attentional cascade for fast rejection of non-face sub-windows**

Training may take a long time

[1] P. A. Viola, M. J. Jones. Rapid object detection using a boosted cascade of simple features, CVPR, pp.511-518, 2001
[2] P. A. Viola, M. J. Jones. Robust Real-Time Face Detection. IJCV 57(2), pp. 137-154, 2004

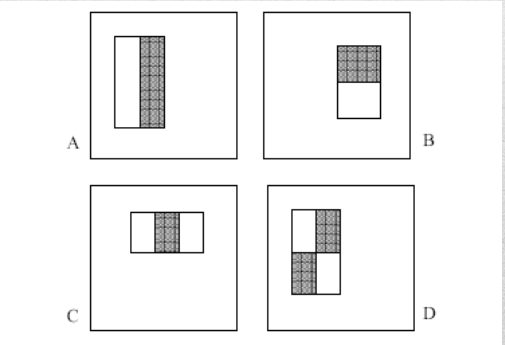
Viola and Jones Steps

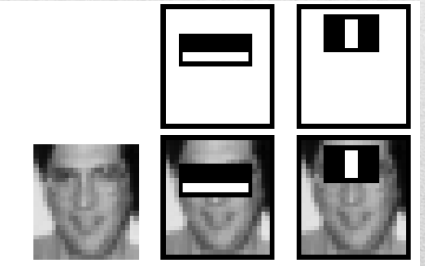
1. **Feature Computation**
   - **Quick Feature Computation**
     - Rectangle features
     - Integral image representation
2. **Feature Selection**
   - **Efficient classification**
     - Ada-Boost training algorithm
3. **Detection (in real time)**
   - **Real-timeliness**
     - A cascade of classifiers

Features

Four basic types

- Easy to calculate
- White areas are subtracted from the black ones
- Integral image representation makes feature extraction faster

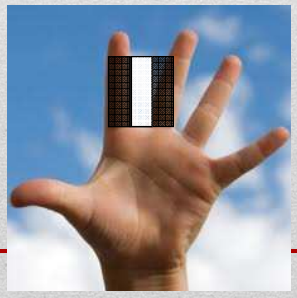




Rectangle Features


Value = ∑ (pixels in white area) – ∑ (pixels in black area)

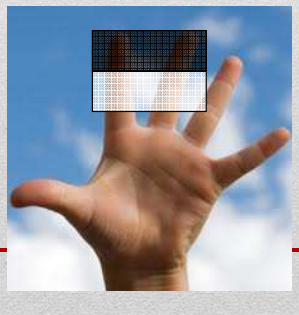




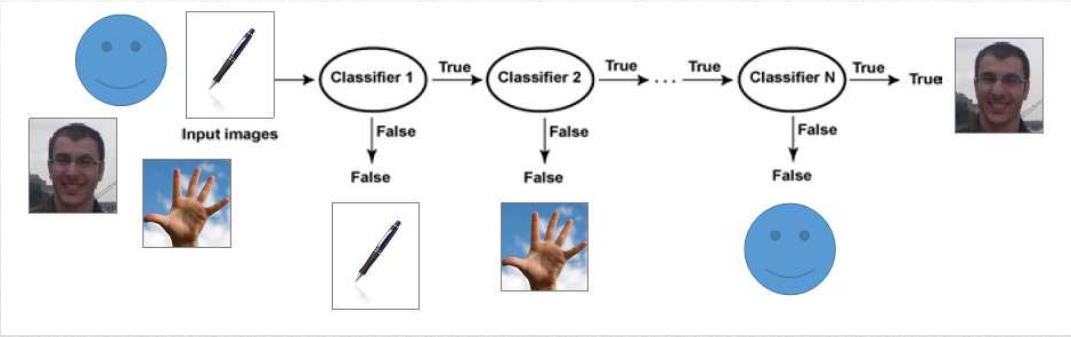

Rectangle Features


Value = ∑ (pixels in white area) – ∑ (pixels in black area)






Cascade of classifiers



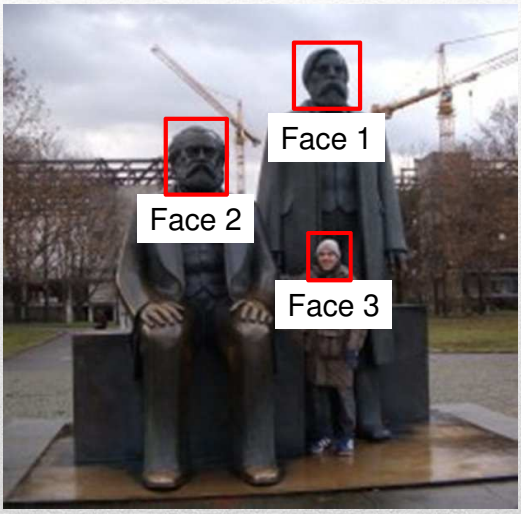
Each classifier "specialized" on rejecting negative samples.

Readings

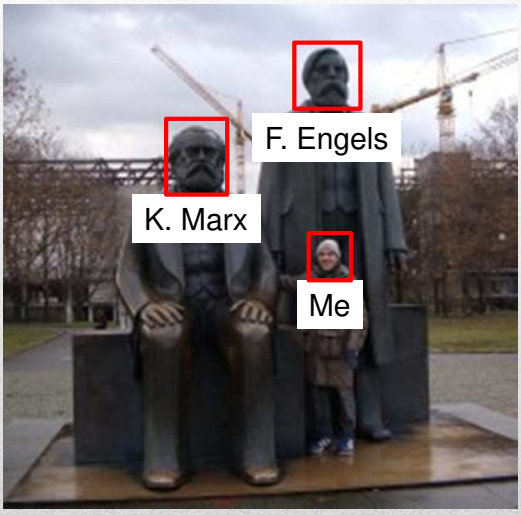
P. Viola and M.J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.

P. Viola and M.J. Jones. Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade. Advances in Neural Information Processing System, pp. 1311-1318, 2001

Detection vs. Recognition

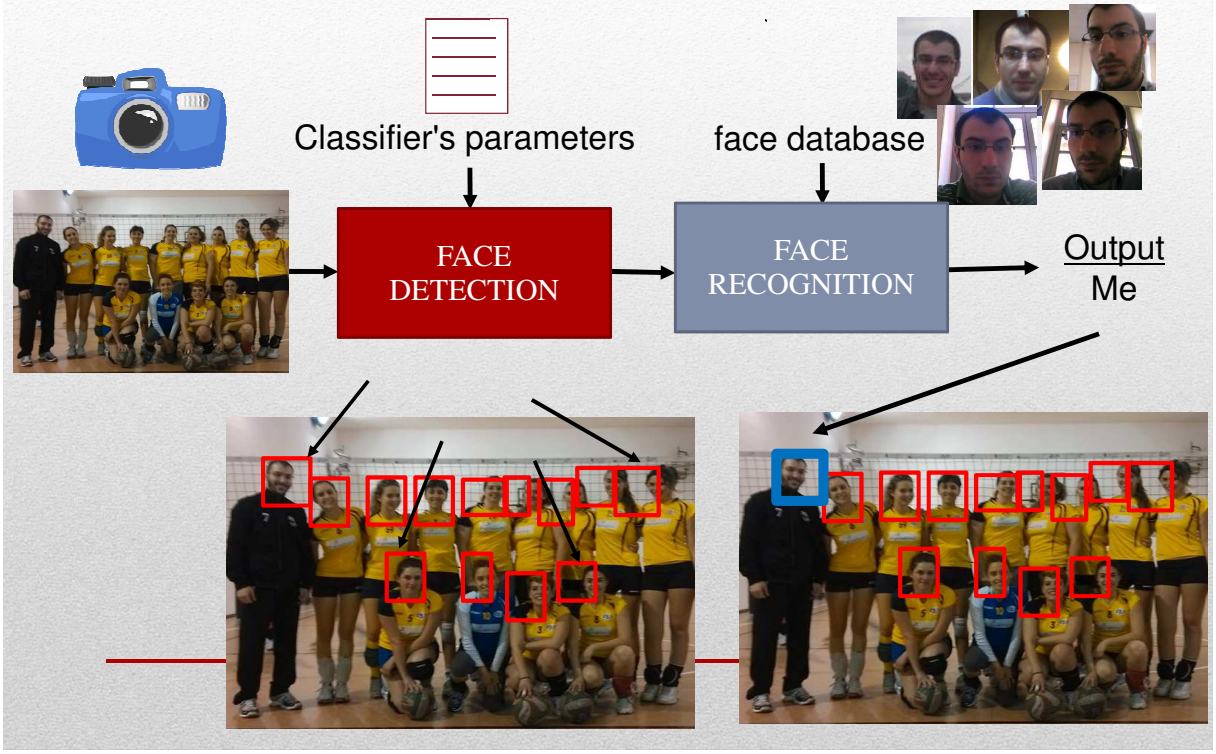


detection



recognition

Detection and Recognition



Face Recognition

Image Features

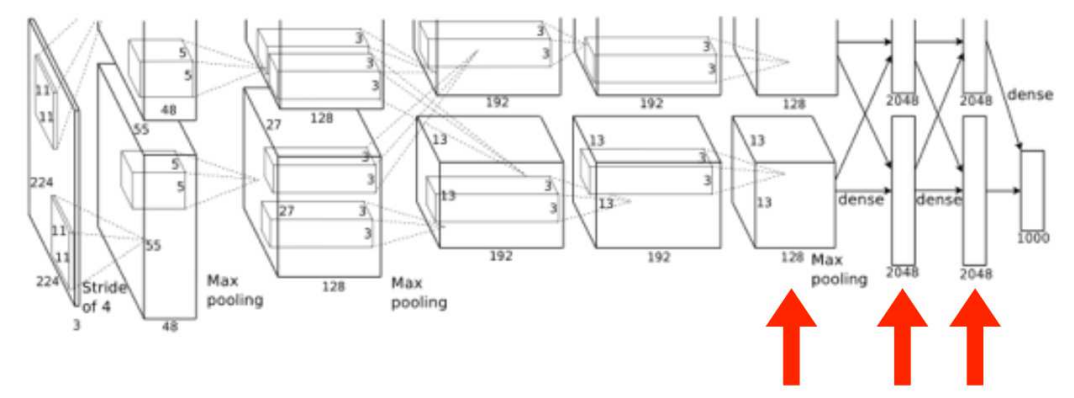
Global features

L

DEEP LEARNING!

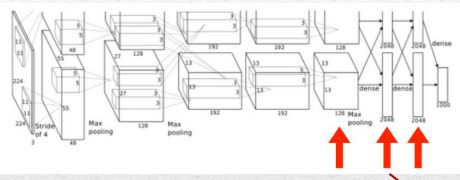
- Scale invariant (SIFT, SURF, …)
- Binary (BRIEF, ORB, …)

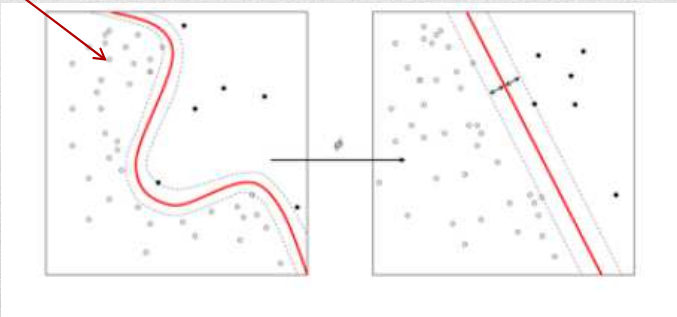
CNN



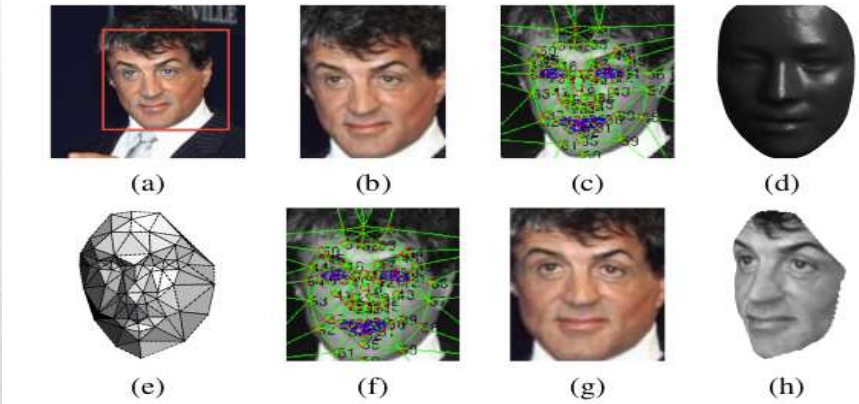
Deep Features

CNN + SVM



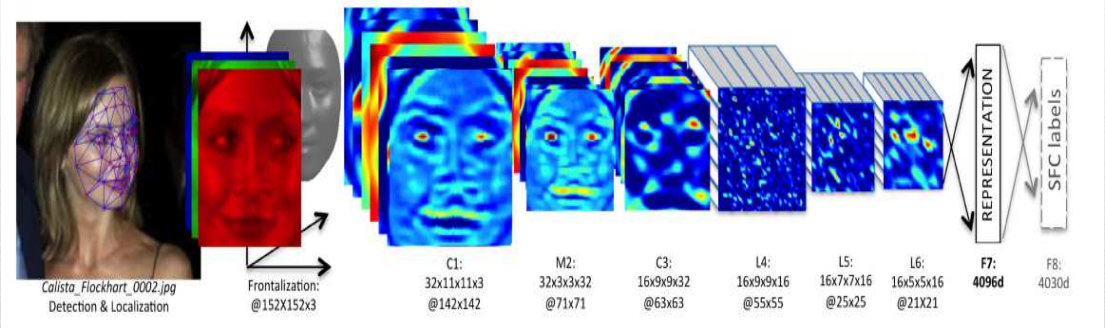


Face Alignment



Deep Face

Deep Face



DeepFace: Closing the Gap to Human-Level Performance in Face Verification

Open Face

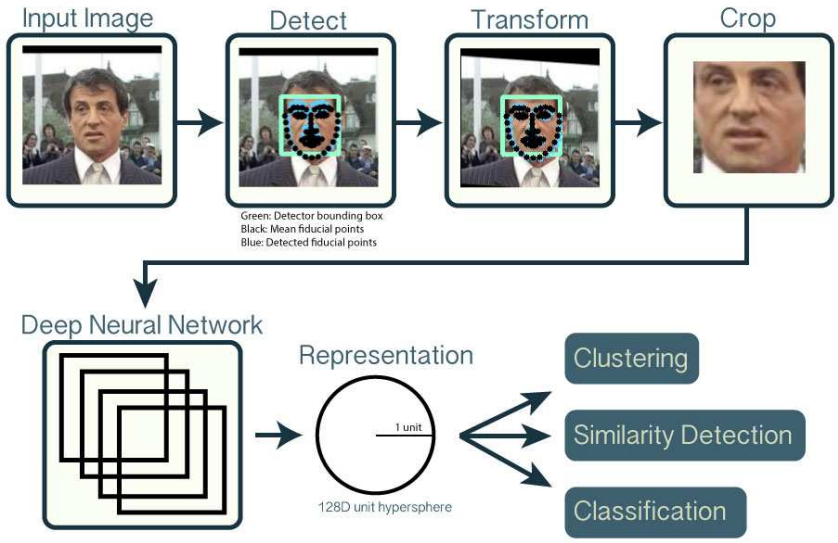
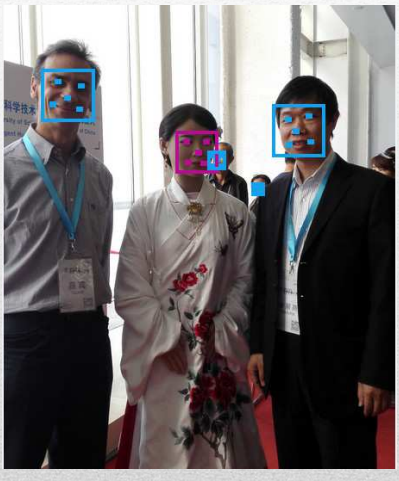


Image from Open Face web ste

Microsoft Cognitive Services

Face API

- Detection
- Verification
- Identification
- Similarities
- Grouping



https://azure.microsoft.com/services/cognitive-services/face/

Microsoft Cognitive Services

- age
- gender
- smile (smile intensity [0,1]).
- facialHair (moustache, beard and sideburns [0,1])
- headPose 3-D roll/yaw/pitch angles
- glasses (glasses type: 'NoGlasses', 'ReadingGlasses', 'Sunglasses', 'SwimmingGoggles'….)
- emotion: emotion intensity, including neutral, anger, contempt, disgust, fear, happiness, sadness and surprise.
- hair: hair values: bald, hair color…
- makeup: whether eye, lip areas are made-up or not.
- accessories: accessories around face, including 'headwear', 'glasses' and 'mask'.
- blur
- exposure
- noise

Readings

Y. Taigman, M. Yang, M. Ranzato, L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in IEEE Conference on Computer Vision and Pattern Recognition pp. 1701-1708, 2014

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proc. of CVPR 2015.

Image Segmentation

Pixel-wise classification → assign a semantic label to each pixel in an image

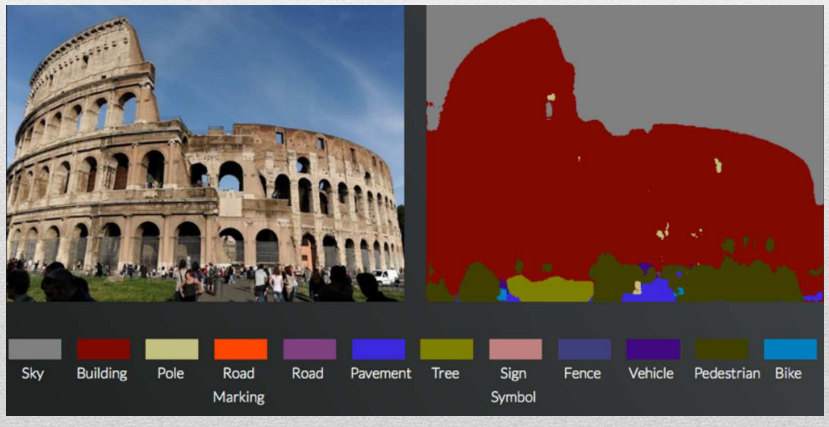


Image Segmentation

Super-pixels: pixels gropued together according to semantic coherence

- Unsupervised
- Efficient processing

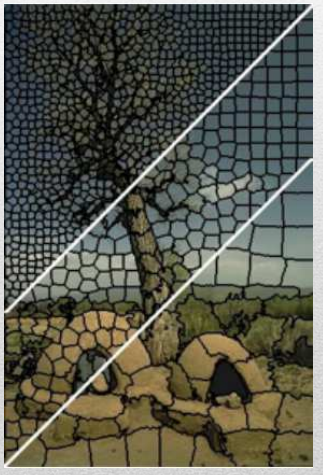


Image Segmentation

Super-pixels + descriptors + semantics

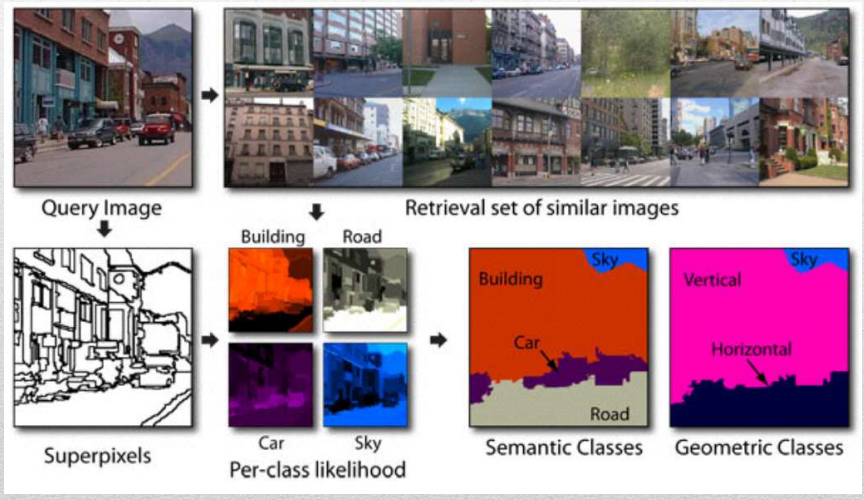
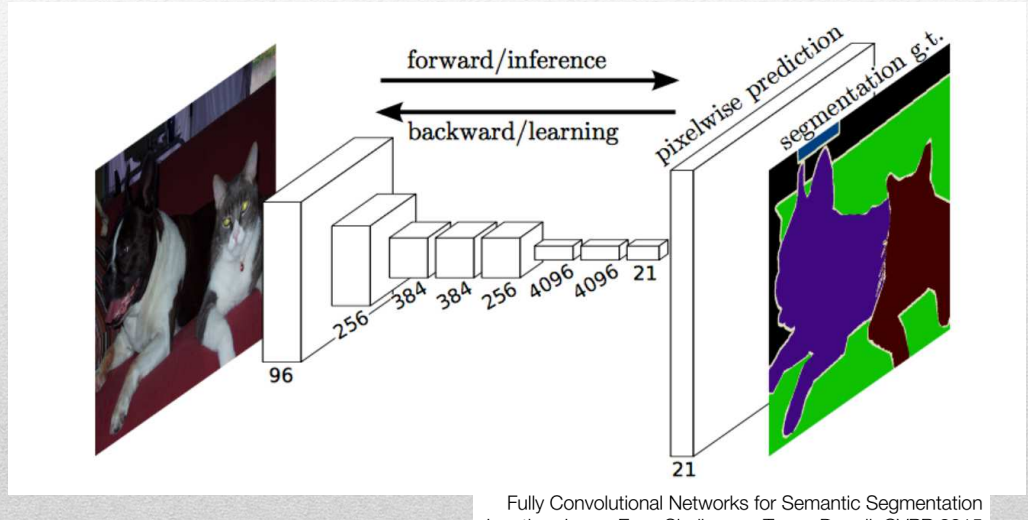


Image Segmentation

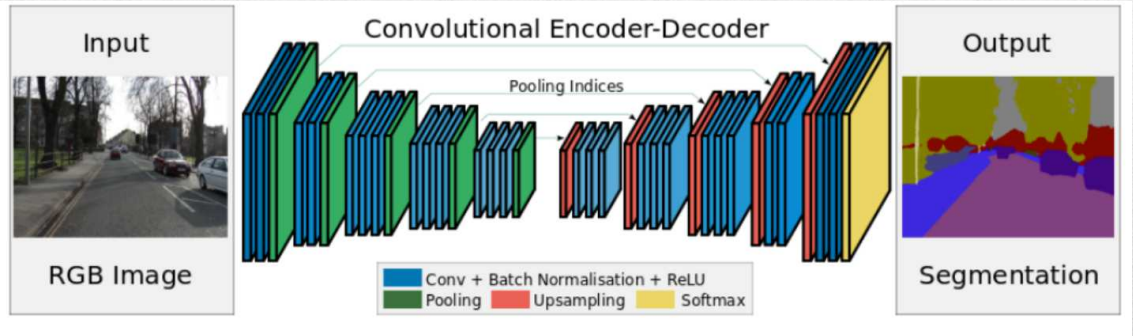
Deep learning



Fully Convolutional Networks for Semantic Segmentation
Jonathan Long, Evan Shelhamer, Trevor Darrell. CVPR 2015

Image Segmentation

Deep learning



Vijay Badrinarayanan, Ankur Handa and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." CVPR, 2015.

Readings

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, CVPR 2014.

Jonathan Long, Evan Shelhamer, Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. CVPR 2015

Vijay Badrinarayanan, Ankur Handa and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." CVPR, 2015.

Credits

Some slides of this presentation adapted from:

- P. Sermanet, “Object Detection with Deep Learning”
- K.H. Wong. “Ch. 6: Face detection”
- P. Viola and T.-W. Yue. “Adaboost for Face Detection”
- D. Miller. “Face Detection & Synthesis using 3D Models & OpenCV”
- S. Lazebnik. “Face detection”
- C. Schmid. “Category-level localization”
- C. Huang and F. Vahid. “Scalable Object Detection Accelerators on FPGAs Using Custom Design Space Exploration”
- P. Smyth. “Face Detection using the Viola-Jones Method”
- K. Palla and A. Kalaitzis. “Robust Real-time Face Detection”
- Images gathered on teh web

Human-Computer Interaction

RGBD Perception

Luca Iocchi

DIAG, Sapienza University of Rome, Italy

With contributions from A. Youssef and M.T. Lazaro

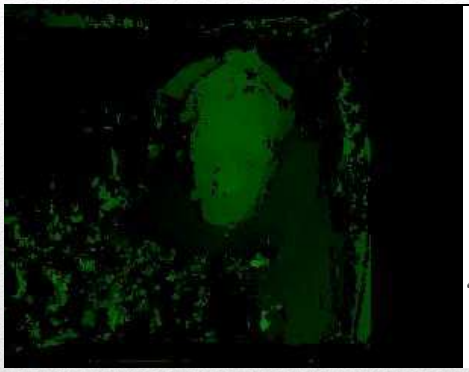
Outline

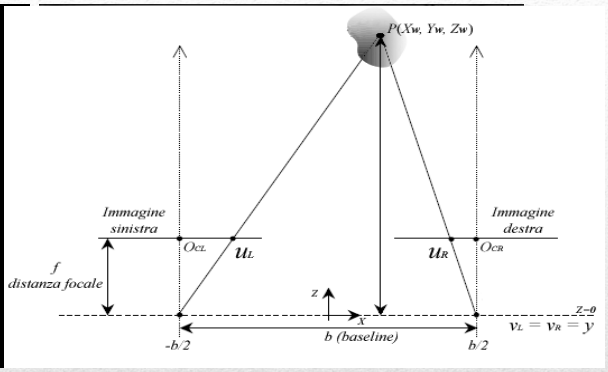
- RGBD sensors and applications
- Detectors for people detection
- RGB processing / OpenCV
- Example 1: RBG face/body detection
- ROS
- Depth processing
- Example 2: RGBD face detections / depth segmentation / virtual buttons
- Conclusions

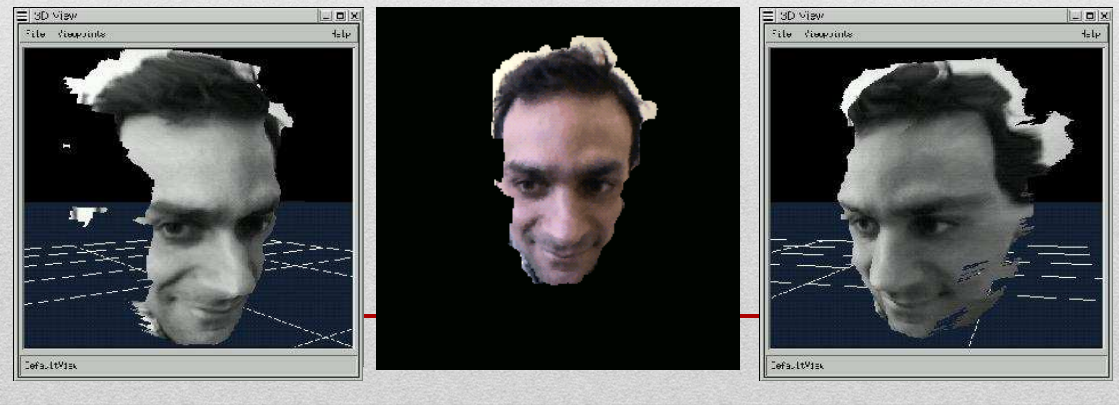
Depth cameras

- Color (RGB) + Depth (D) information
- Improve efficiency and robustness of image processing
- Mostly used in video-games, but useful also in HCI and HRI

Stereo vision

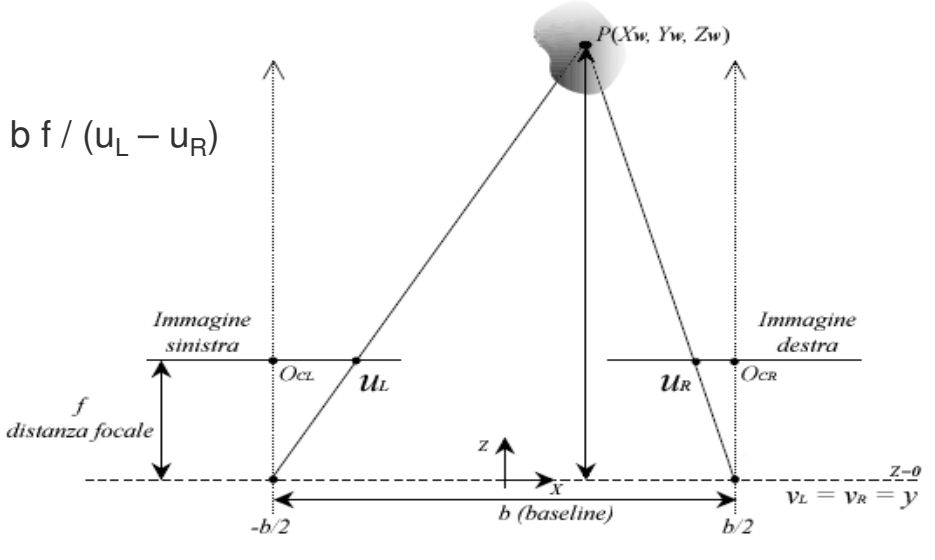






Stereo Triangulation

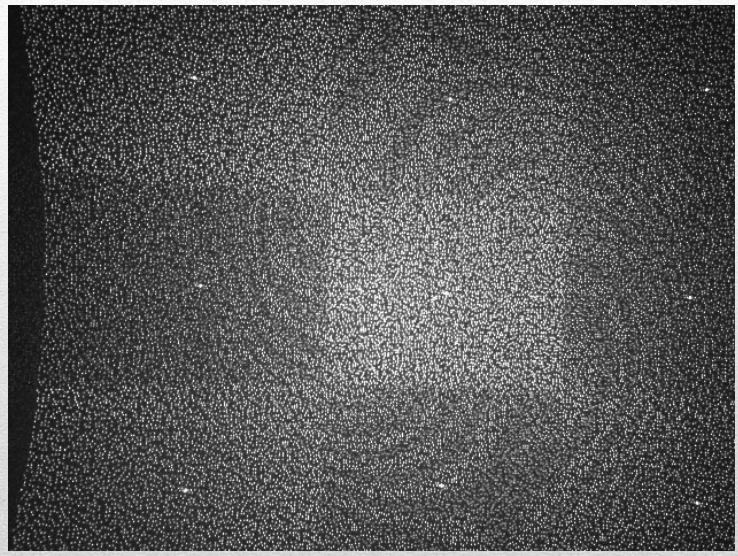
$Z = b\, f / (u_L - u_R)$



Active RGBD cameras

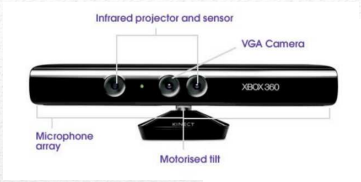
- Capture color and depth
- Active infra-red light pattern
- Work with poor/no texture
- Depth computation
  - Stereo triangulation
  - Time of flight
- Indoor (dark) environments

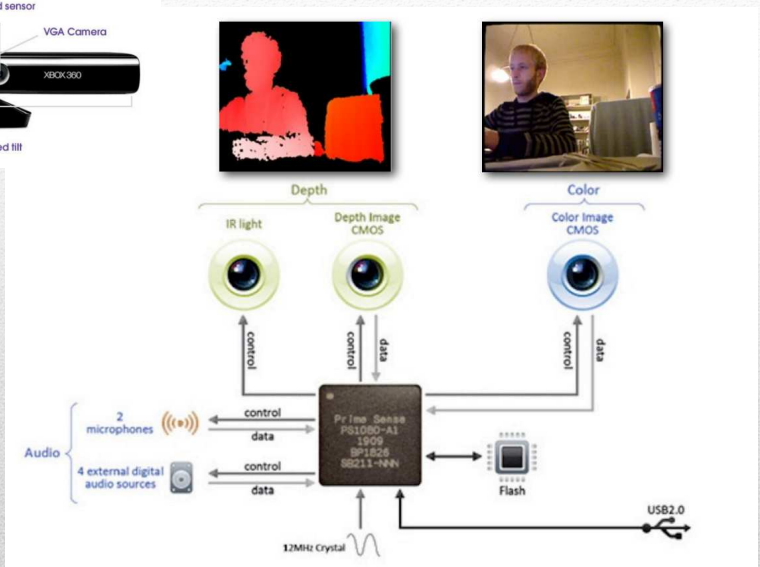
Active infra-red pattern



http://wiki.ros.org/kinect_calibration/technical

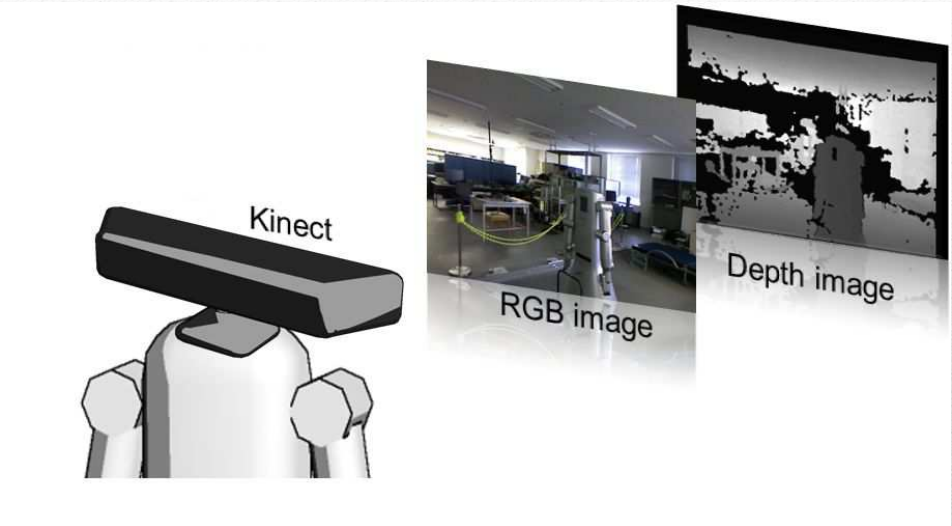
RGBD Vision





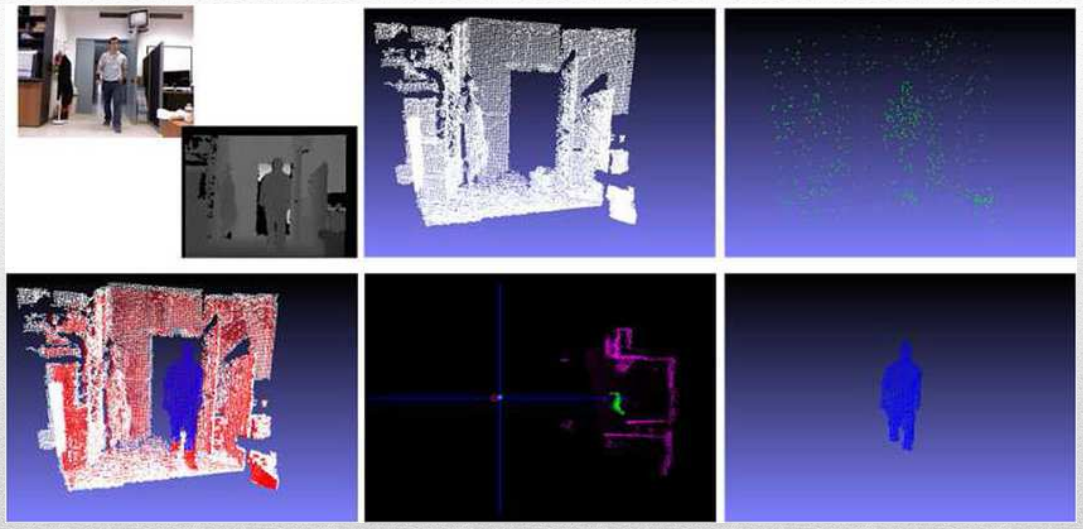
Kinect and RGBD Images: Challenges and Applications. Luiz Velho. IMPA

RGBD Vision



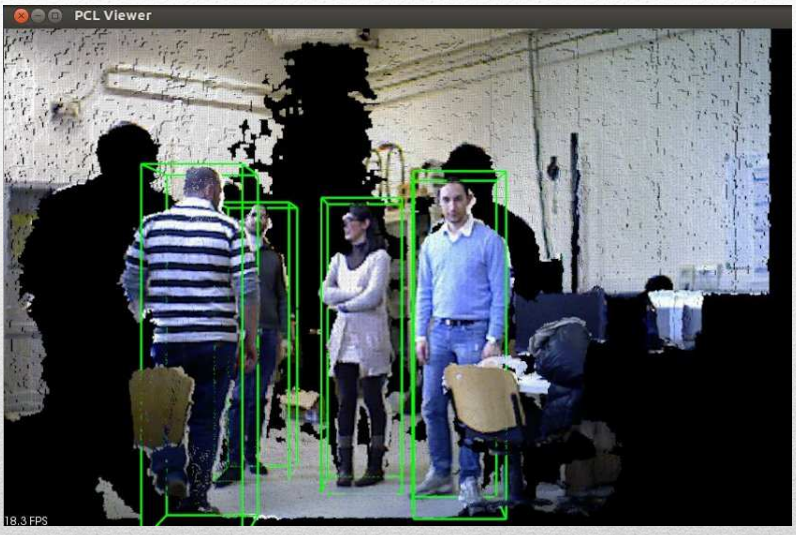
http://robotics.ait.kyushu-u.ac.jp/~kurazume/

RGBD segmentation



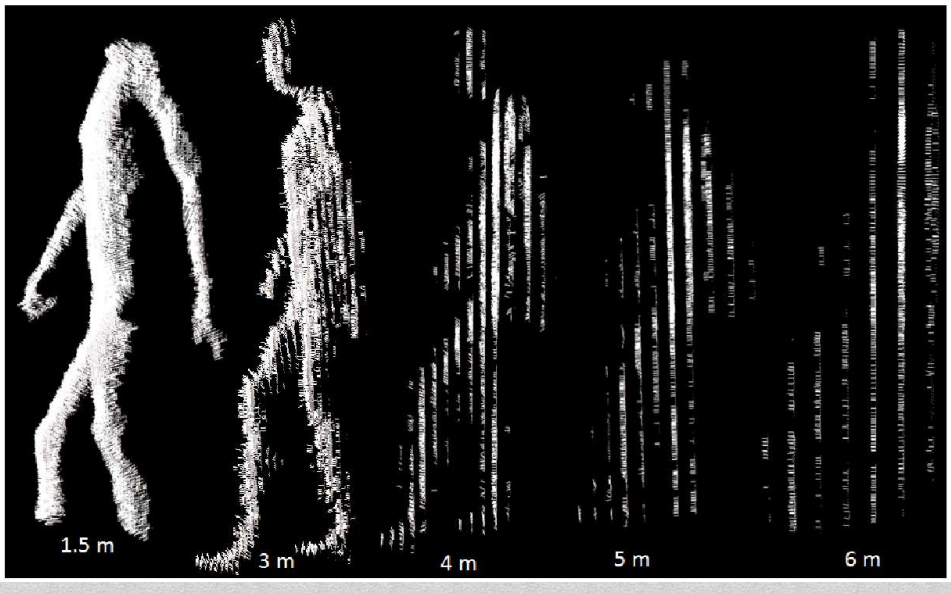
Palopoli et al. Navigation assistance and guidance of older adults across complex public spaces: the DALi approach. Intelligent Service Robotics 8(2):77-92 · April 2015

Point Clouds



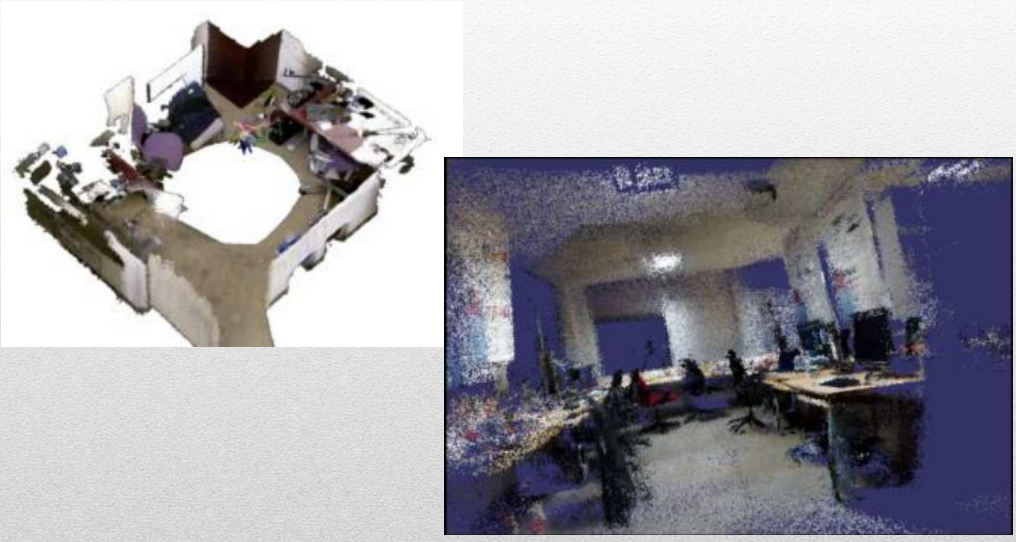
http://pointclouds.org/documentation/tutorials/ground_based_rgbd_people_detection.php

Depth resolution



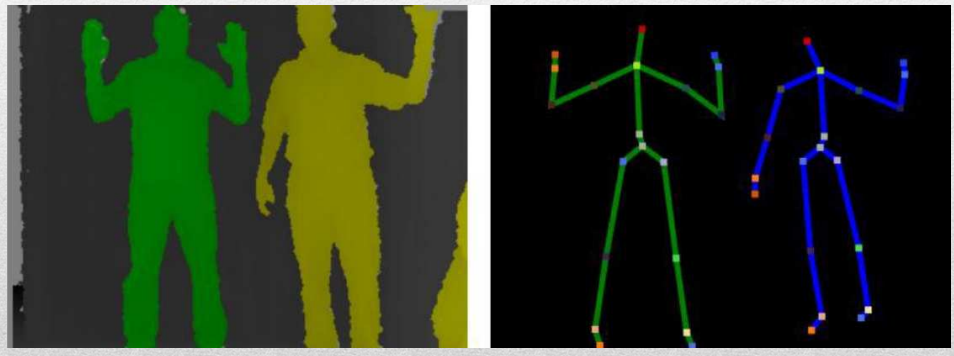
Krystof Litomisky Consumer RGB-D Cameras and their Applications

Application: 3D mapping



Krystof Litomisky Consumer RGB-D Cameras and their Applications

Application: skeleton tracking

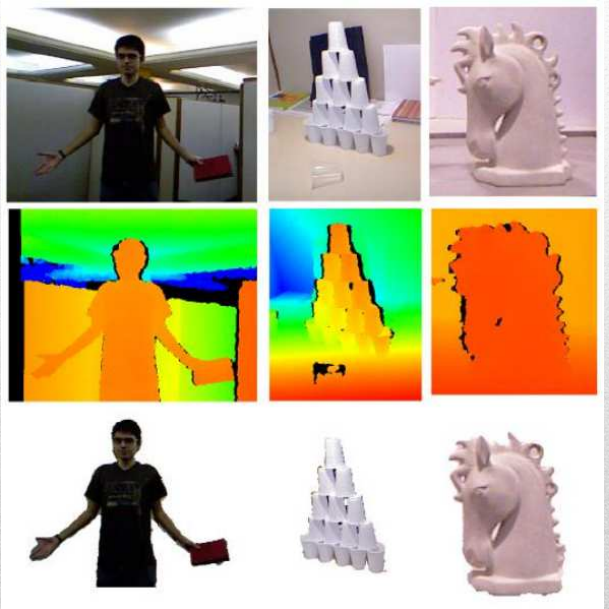


SkeletalViewer, Microsft Kinect SDK

Application: augmented reality

SkeletalViewer, Microsft Kinect SDK

RGBD Image processing



Efficiency and robustness

Depth segmentation

Image Processing

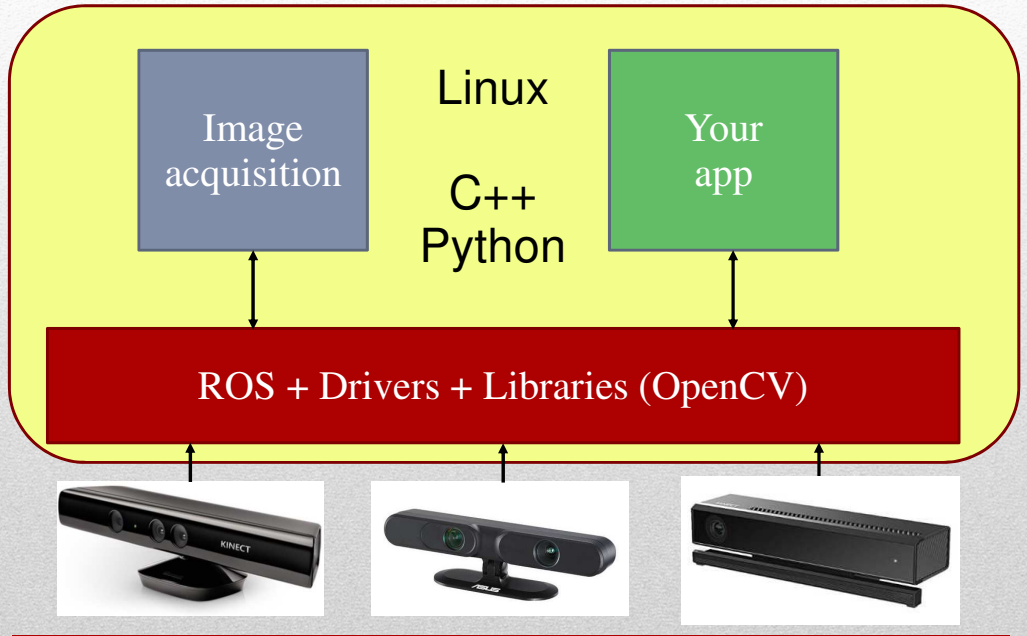
Depth filtering

OUTPUT

Select region of interest

Remove false positives

Software Libraries



Software Installation

Some effort needed … but we can provide

- Docker containers
- Image for Raspberry PI 3

Introduction to OpenCV

- OpenCV (Open Source Computer Vision) is a library of programming functions for realtime computer vision.
- BSD Licensed
- free for commercial use
- C++, C, Python and Java (Android) interfaces
- Supports Windows, Linux, Android, iOS and Mac OS
- More than 2500 optimized algorithms

http://opencv.org/

Introduction to OpenCV

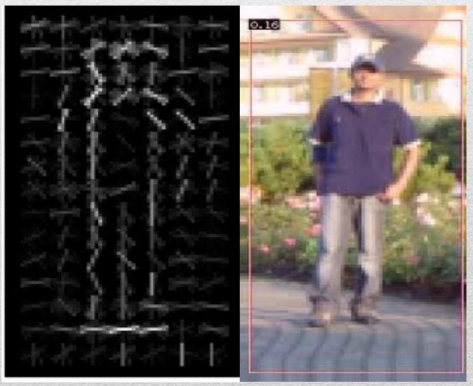
Modules for Image Processing

- **core** - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
- **imgproc** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- **features2d** - salient feature detectors, descriptors, and descriptor matchers.
- **highgui** - an easy-to-use interface to video capturing, image and video codecs, as well as simple UI capabilities.

Full body detection in images

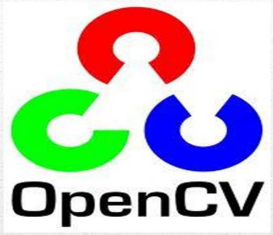
Histogram of Oriented Gradient (HOG)

- It was introduced by Navneed Dalal and Bill Triggs in 2005 [1]
- Sliding window technique for people detection in image.
- Shape and appearance presence.
- HOG is a features descriptor:
  - Dense feature extraction.
  - Local overlapping.
  - Trained classifier (support Vector Machine SVM)



Full body detection in images

HOG in OpenCV (C++)




```
#include <opencv2/objdetect/objdetect.hpp>

HOGDescriptor hog; // standard descriptor

hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());

vector<Rect> found; // where to save the detected persons

hog.detectMultiScale(img, found, 0, Size(8,8), Size(32,32), 1.05, 2);
```

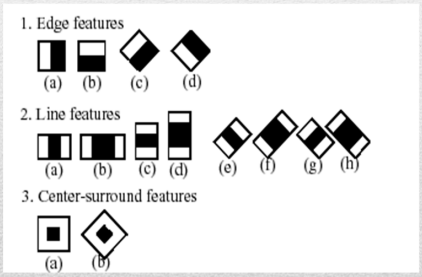
http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

Face detection in images

Viola-Jones implementation of the OpenCV library (by using Haar-like cascades)

OpenCV comes with a trainer as well as a detector
OpenCV already contains many pre-trained classifiers for face, eyes, smile etc.

Cascade Classifier in OpenCV



C++

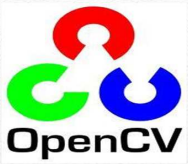
```
void CascadeClassifier::detectMultiScale
      (const Mat& image,
      vector<Rect>& objects,
      double scaleFactor=1.1,
      int minNeighbors=3,
      int flags=0,
      Size minSize=Size(), Size maxSize=Size() )
```

Face detection in images

Cascade Classifier (C++)

```
#include <opencv2/objdetect/objdetect.hpp>

String face_cascade_trained ="haarcascade_frontalface_alt.xml";

CascadeClassifier face_cascade;

face_cascade.load( face_cascade_trained );

vector<Rect> faces;

face_cascade.detectMultiScale( frame_gray, faces, 1.1, 2,
            0|CV_HAAR_SCALE_IMAGE, Size(30, 30) );
```


Face detection in images

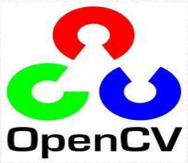


Cascade Classifier (Python)

```
import cv2

facemodel = '…/opencv/haarcascades/haarcascade_frontalface_default.xml'
faceCascade = cv2.CascadeClassifier(facemodel)

grayimg = …

faces = faceCascade.detectMultiScale(grayimg, scaleFactor=1.1,
      minNeighbors=5, minSize=(30, 30) )

 for (x, y, w, h) in faces:
          cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

OpenCV Classifiers

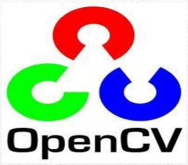
Pre-trained classifiers.




```
CascadeClassifier  detector;

detector.load(TrainedClassifier.xml);

detector.detectMultiScale( Mat& image, vector<Rect>& results,
      1.1, 3, 0|CV_HAAR_SCALE_IMAGE, Size(30, 30)) ;

for (int i=0; i<results.size(); i++) {
      ....
}
```


OpenCV Classifiers

Pre-trained classifiers available in OpenCV



haarcascade_frontalface_alt.xml
haarcascade_eye_tree_eyeglasses.xml
haarcascade_mcs_leftear.xml
haarcascade_mcs_rightear.xml
haarcascade_mcs_mouth.xml
haarcascade_mcs_nose.xml
haarcascade_smile.xml
haarcascade_upperbody.xml
haarcascade_lowerbody.xml
hogcascade_pedestrians.xml
haarcascade_fullbody.xml

Robot Operating System - ROS

- ROS is a middleware for efficient data exchange
- Based on publish/subscribe and event-based paradigms
- Topics identified by name and type of message
- Each node can publish topics and subscribe to topics
- All nodes subscribed to a topic are notified when a node publishes data on such topic

ROS publish/subscribe

In our examples, camera nodes publish data (i.e., images) on topics whenever they are captured from the devices.

Application nodes subscribe to these topics and are notified when images are ready.

Application nodes are implemented as callback functions activated upon arrival of data in a topic.

ROS/OpenCV link

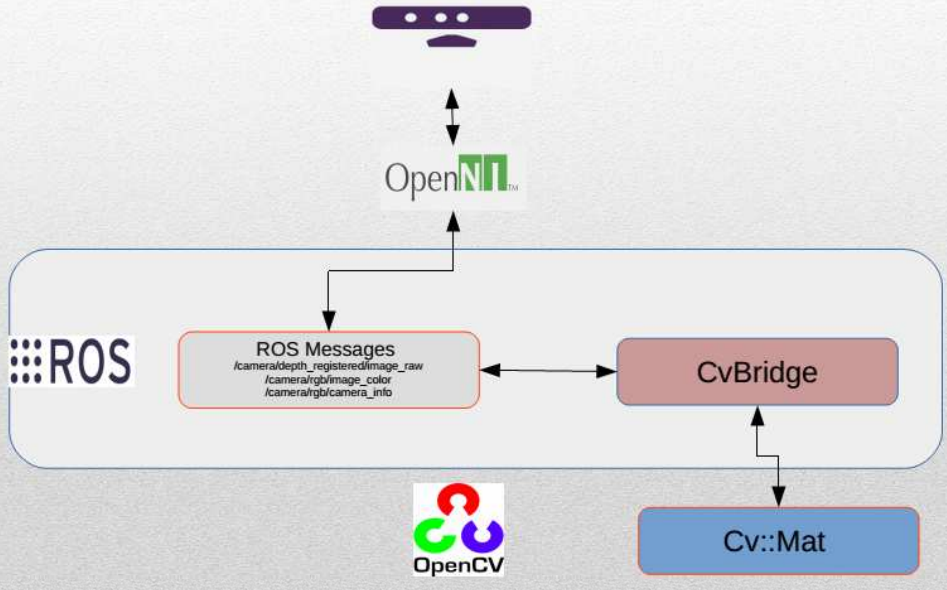


Image processing

```
class Processor

void rgb_proc( … ) {

      cv::Mat  image = cv_rgb_ptr->image;
      ….

}

void depth_proc( … ) {

      cv::Mat  image = cv_depth_ptr->image;

      short z = depthImage.at<short>(i, j); // depth in mm
}
```

Example

RGBD face/person detection, depth segmentation, virtual buttons

https://bitbucket.org/iocchi/rgbd_person_detection

- Processor
- display
- Face/body Detection
- depthSegmentation
- virtualButtons

Conclusions

- RGBD computation improves image processing tasks, specially for person detection
- Many applications relevant for HRI and HCI
- Many ideas for projects
  - Advanced user input through person feature detection and recognition
  - Augmented reality
  - …