PROJECT
Code Implementation
(Object Recognition)

LABORATORY SESSIONS

OBJECTIVES

- Implementation of a Matlab GUI to detect, describe and recognize objects.

© O.A.Brekke 2006 E

STUDENT’S WORK

- Implementation of a Matlab code with a **graphical interface** (GUI or app) aimed to recognize a certain type of objects. Alternatively, any other task related to image processing can be implemented.

- All the required files must be uploaded to PoliformaT:
  - **Matlab files** required to run the developed software. For its evaluation, the code must provide a proper result without errors.
  - **Report** (Word or PDF) with the project documentation.

- Evaluation of the work will take into account the problem’s difficulty, code’s originality, programming structure and comments, and report’s content, quality and presentation.

STUDENT’S WORK

Report Contents

- Introduction: description of the project’s objectives.
- GUI: description of the GUI’s layout and objects.
- Code: the code should be commented, specially the key lines of the process. There is no need to include code lines in the report.
- Description of the algorithm: flowchart of the detection process and some sample images summarizing its performance.
- Results: analysis of the code performance with several examples (successful and failed).
- Discussion of results and possible improvements.
- References: sources of information consulted.

STUDENT'S WORK

Graphical Interface

- The graphical interface in Matlab can be developed by the **GUIDE** tool or the **App Designer**.

- The GUIDE tool is a graphical design environment to develop GUIs (Graphical User Interfaces) in Matlab. This tool will be removed in future Matlab releases.

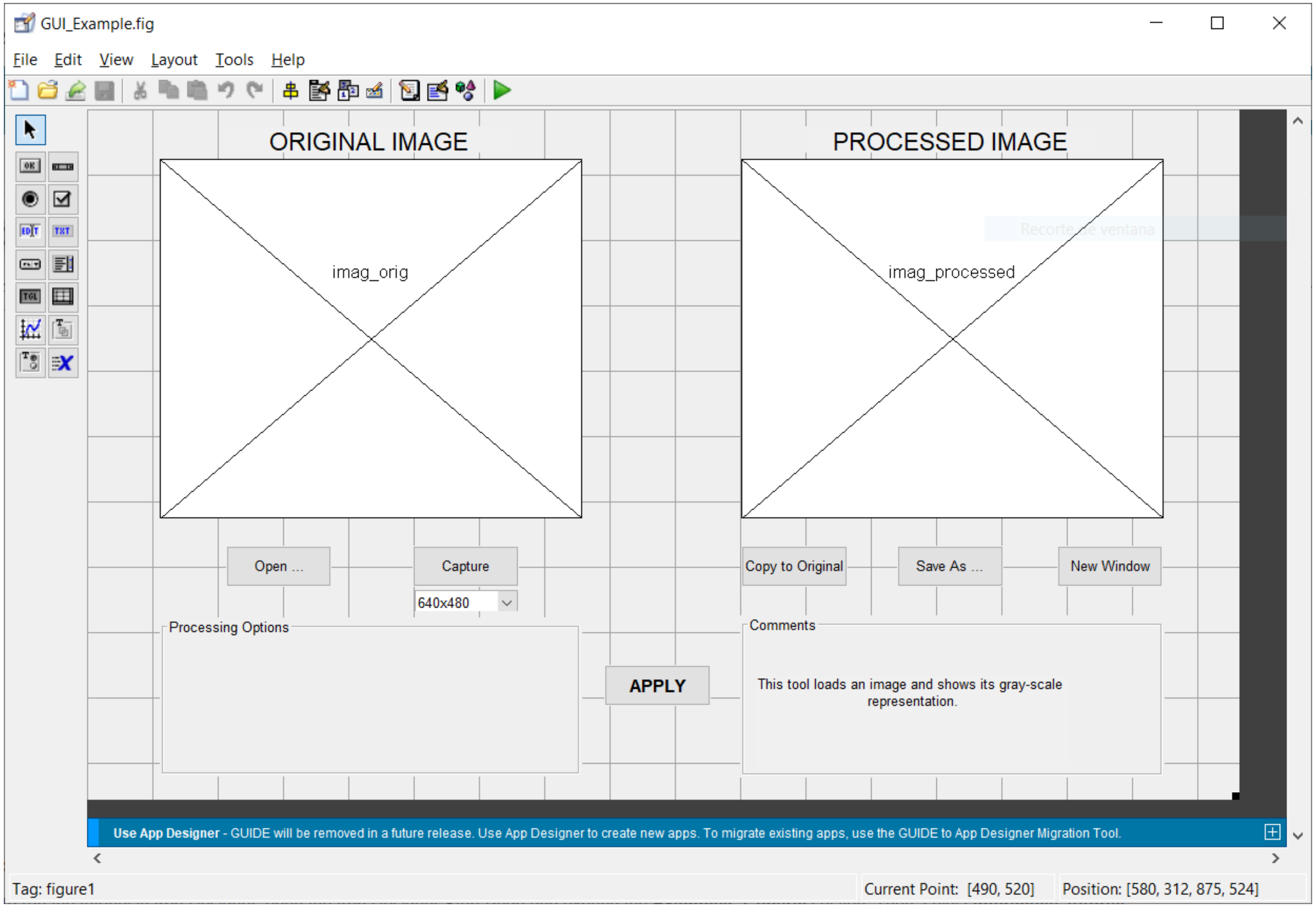
  Create a Simple App Using GUIDE: https://mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html

- The App Designer is the recommended environment to build graphical apps in new releases of Matlab.

  Create and Run a Simple App Using App Designer: https://mathworks.com/help/matlab/creating_guis/create-a-simple-app-or-gui-using-app-designer.html

STUDENT’S WORK

Graphical Interface



This tool loads an image and shows its gray-scale representation.

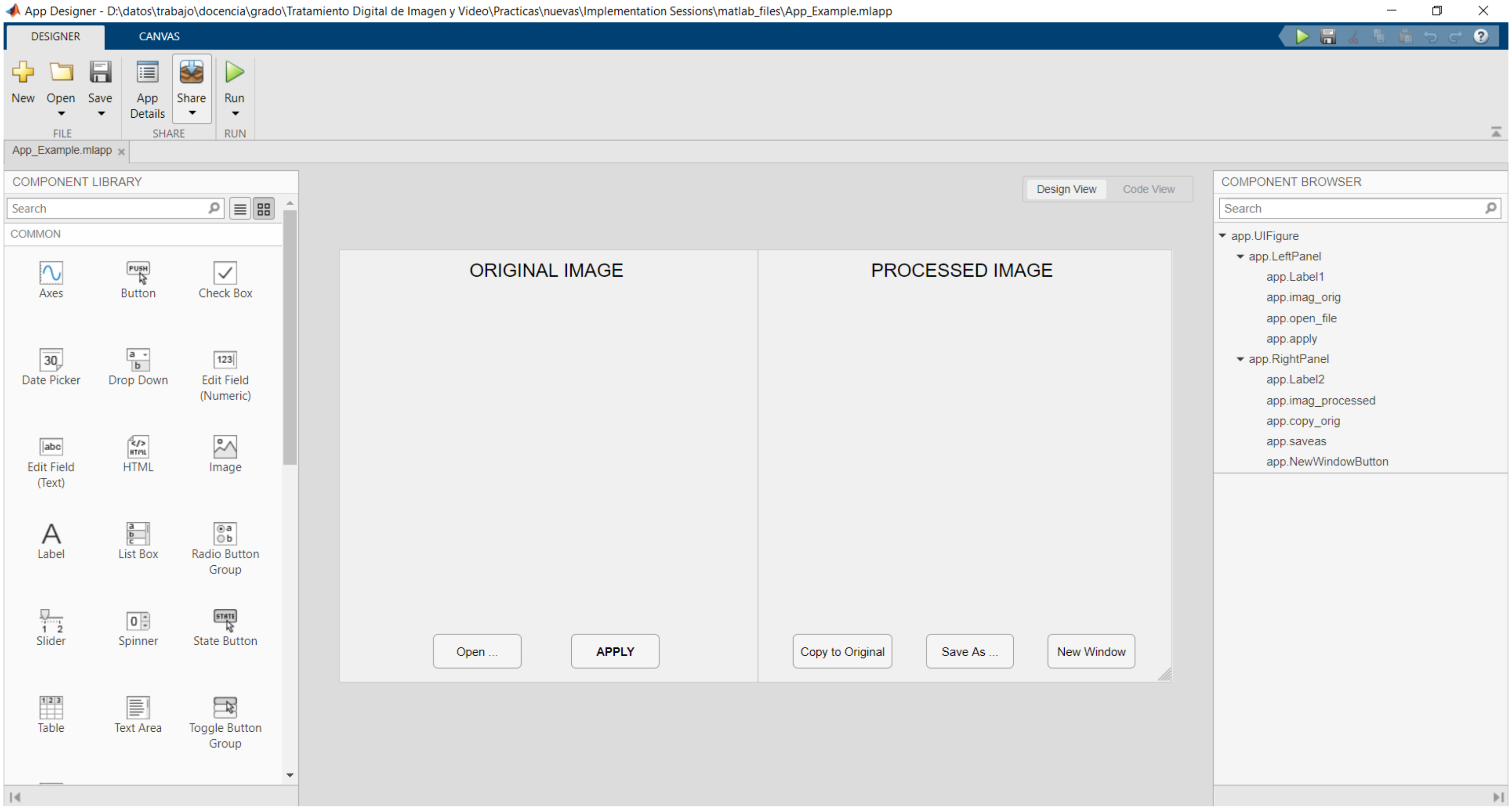
Use App Designer - GUIDE will be removed in a future release. Use App Designer to create new apps. To migrate existing apps, use the GUIDE to App Designer Migration Tool.

Tag: figure1 Current Point: [490, 520] Position: [580, 312, 875, 524]

GUI_Example.fig

STUDENT'S WORK

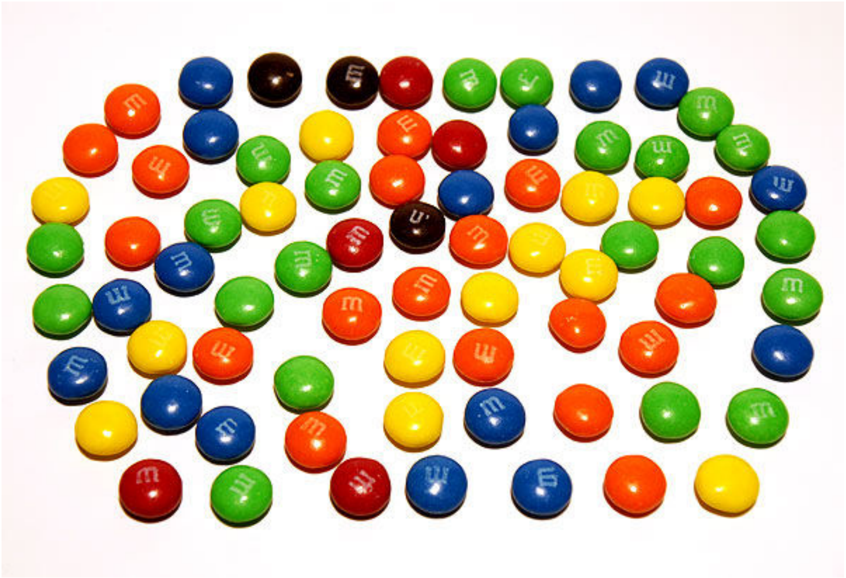
Graphical Interface



App Designer - D:\datos\trabajo\docencia\grado\Tratamiento Digital de Imagen y Video\Practicas\nuevas\Implementation Sessions\matlab_files\App_Example.mlapp

App_Example.mlapp

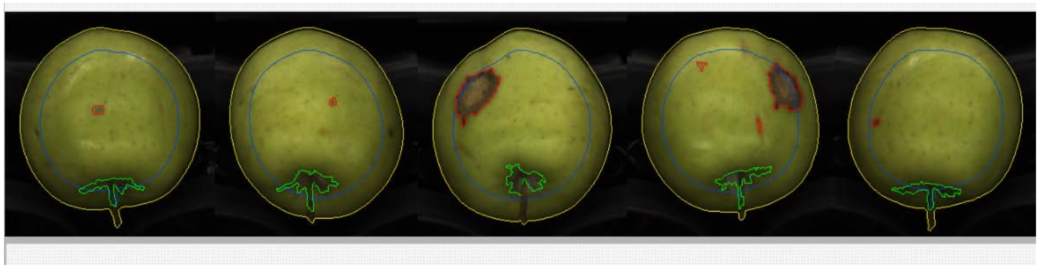
PROJECT IDEAS

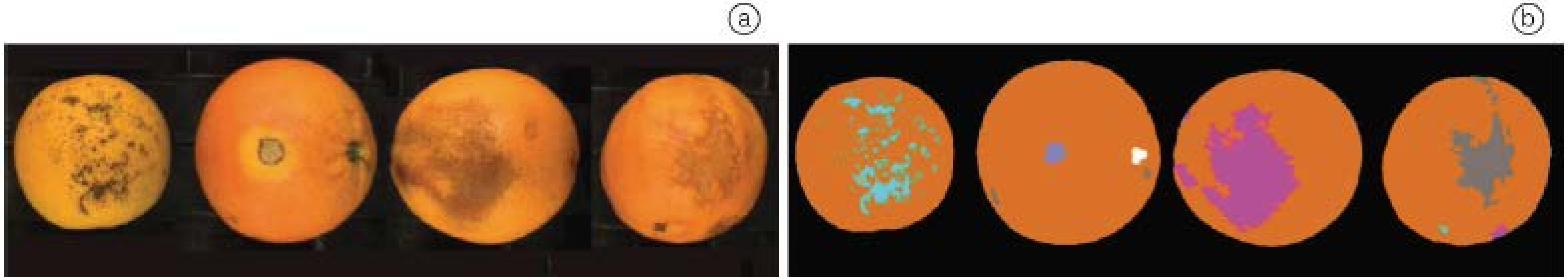


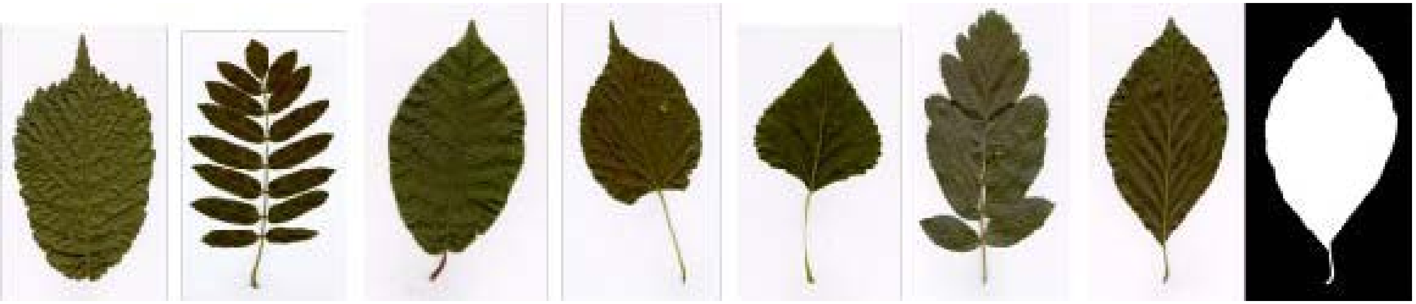
Quality control for industry

PROJECT IDEAS

Leaf/Fruit Recognition

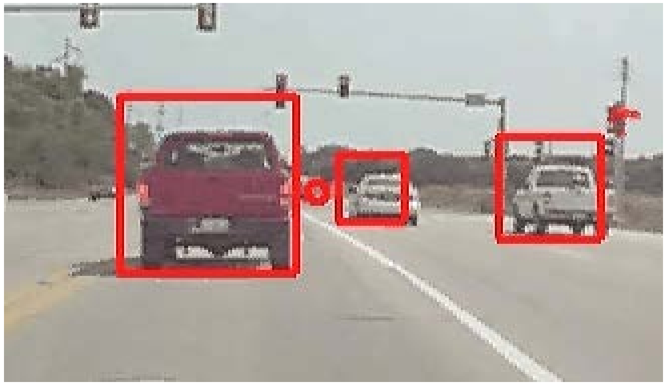


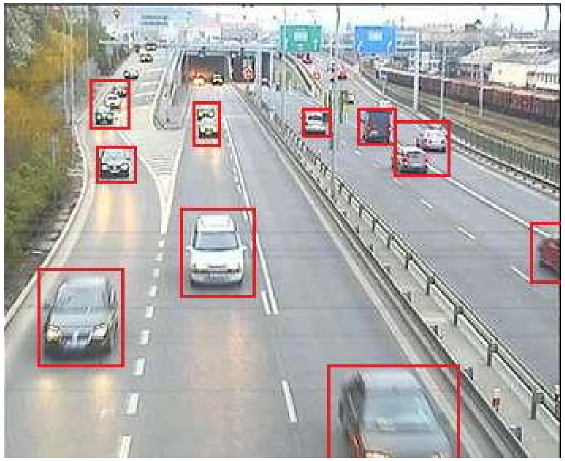


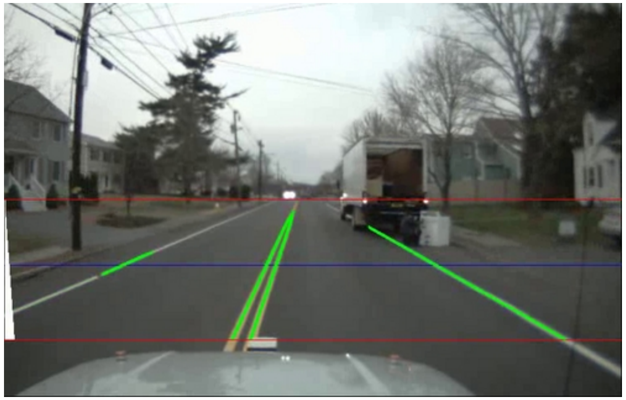


PROJECT IDEAS

Driving assistance

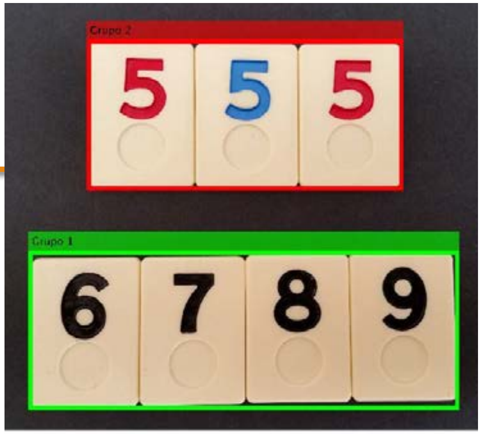




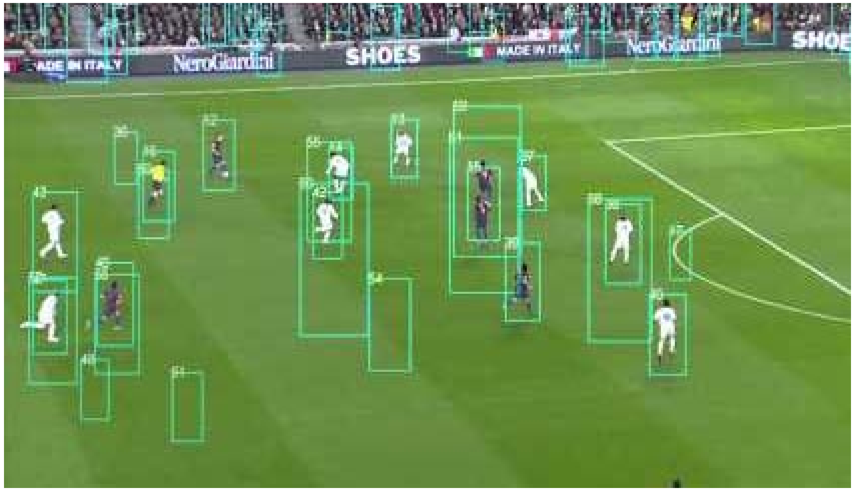


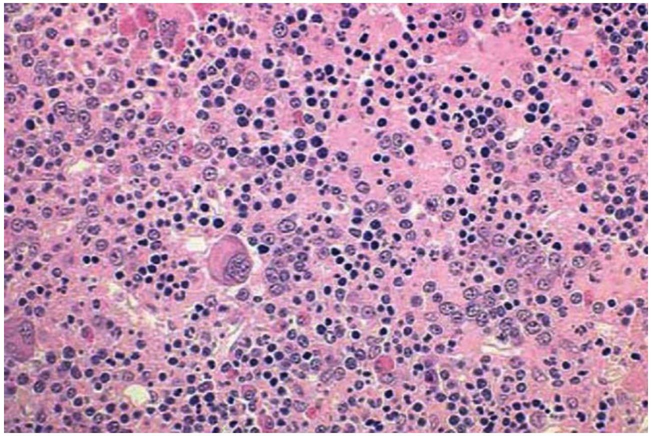
© O.A.Brekke 2006 E

PROJECT IDEAS

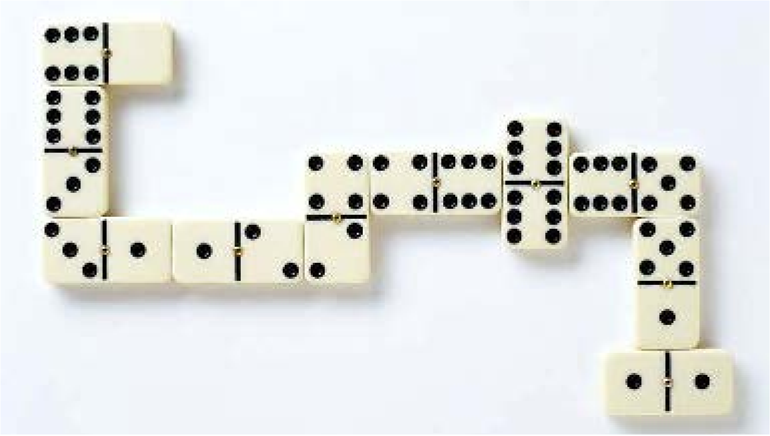


Sports/games



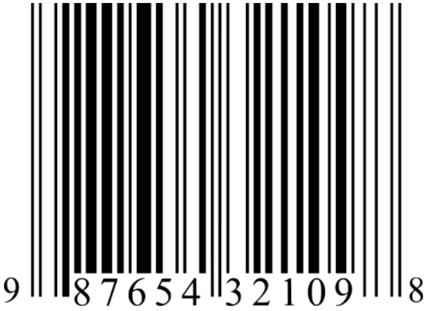


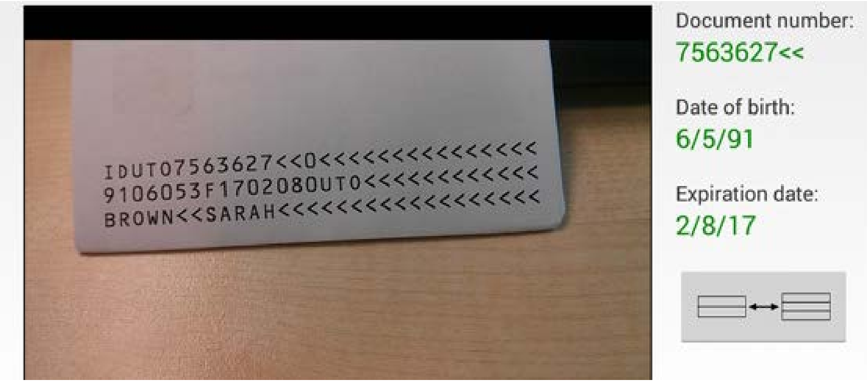
Cell counting



PROJECT IDEAS

Barcode/Card/Paper Reader



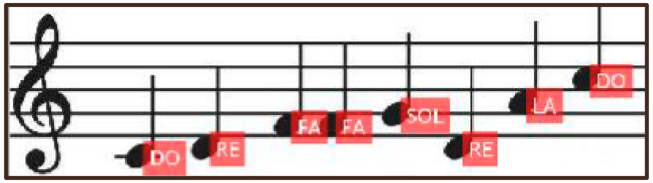


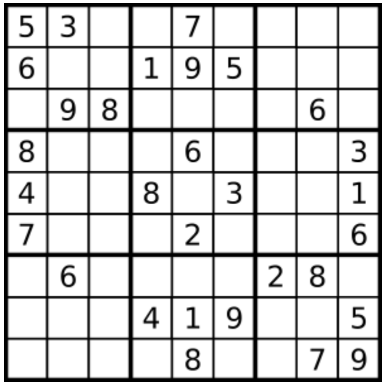
IDUT07563627<<0<<<<<<<<<<<<<<<
9106053F1702080UT0<<<<<<<<<<<
BROWN<<SARAH<<<<<<<<<<<<<<<<<<

Document number:
7563627<<

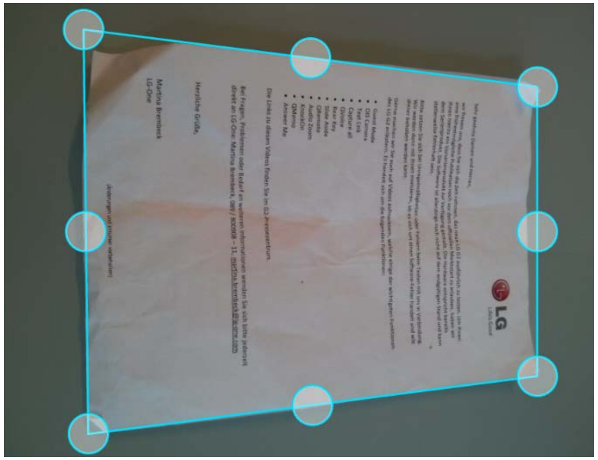
Date of birth:
6/5/91

Expiration date:
2/8/17



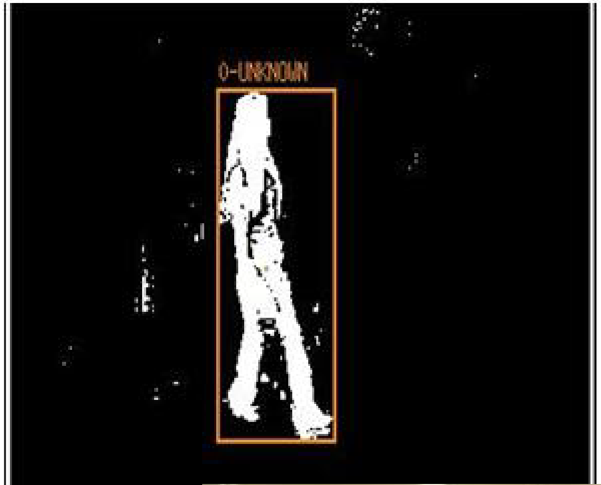


5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9



PROJECT IDEAS

People Detection



CAMERAS

PC webcams:

- Install *MATLAB Support Package for USB Webcams*
- Matlab code to acquire images/video:

```
camList = webcamlist; % show the connected webcams
cam = webcam(1); % connect to the webcam (first one)
preview(cam); % open a video preview window
img = snapshot(cam); % get a frame
imshow(img); % show the captured frame
```

CAMERAS

Android phone:

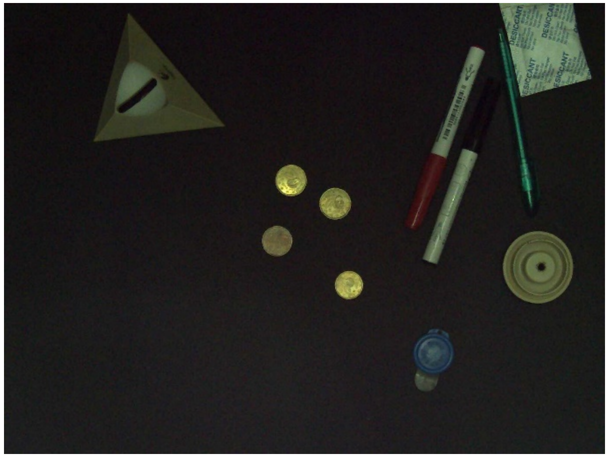
- Install app *IP Webcam* or similar
- Open the app, start the server and write the IP address
- Matlab code to acquire images/video:

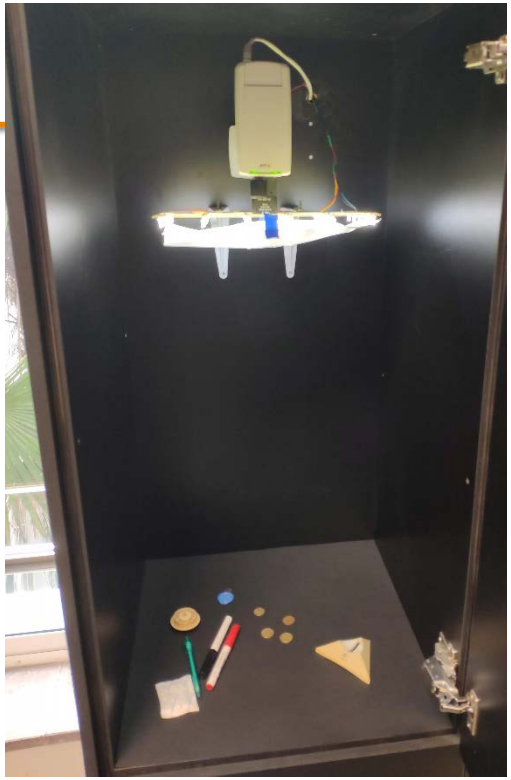
```
img = imread('http://ip_address/shot.jpg’); % read image

cam = ipcam('http://ip_address/video'); % connect to the video stream
preview(cam); % open a video preview window
img = snapshot(cam); % get a frame
```

CAMERAS

Cabinet camera:





- Matlab code to acquire images/video:

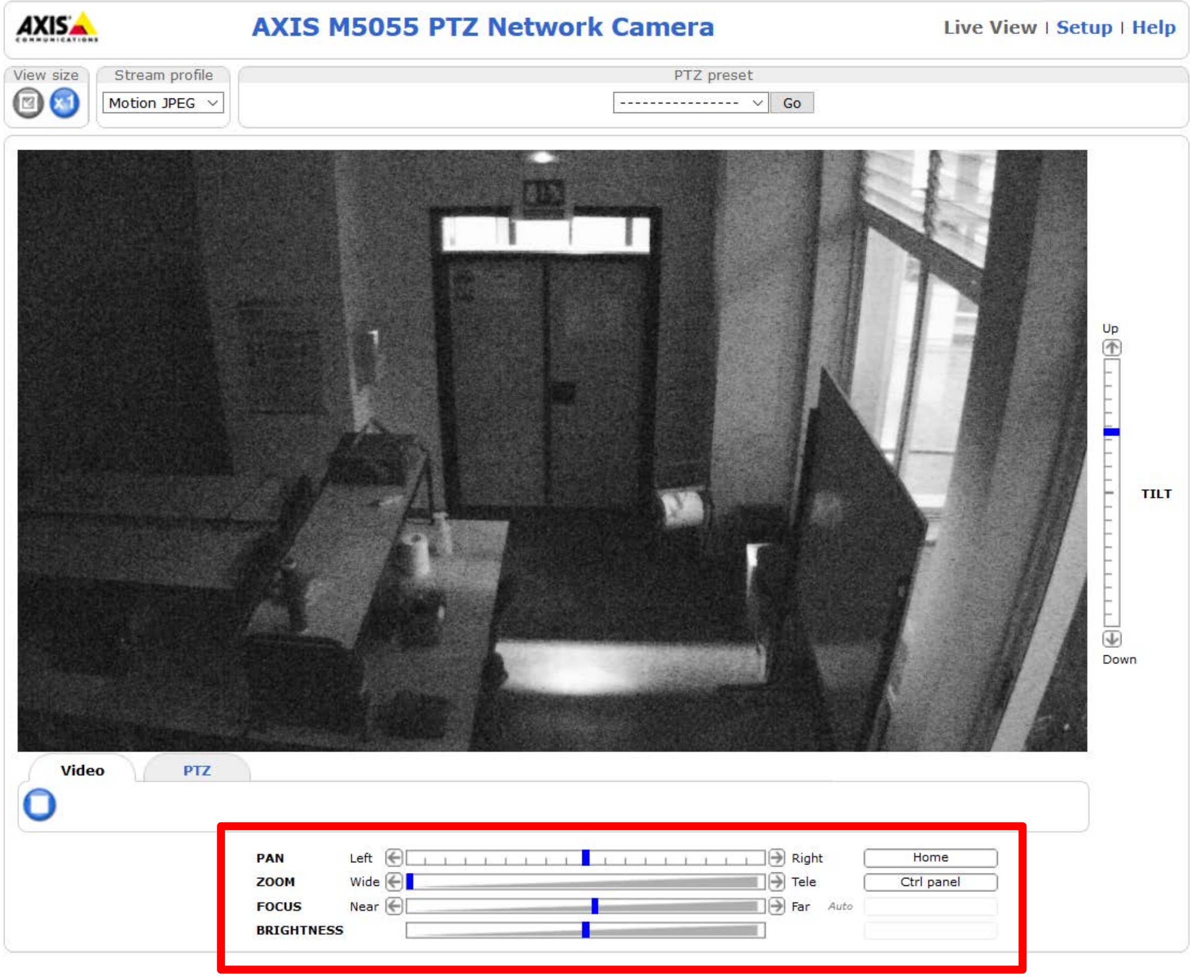
```
img = imread('http://visionartificia.gnd.upv.es/jpg/image.jpg’); % read image

% connect to the video stream
cam = ipcam('http://visionartificia.gnd.upv.es/mjpg/1/video.mjpg');
preview(cam); % open a video preview window
img = snapshot(cam); % get a frame
```

CAMERAS

Lab camera:

PTZ (Pan-Tilt-Zoom)



CAMERAS

Lab camera:

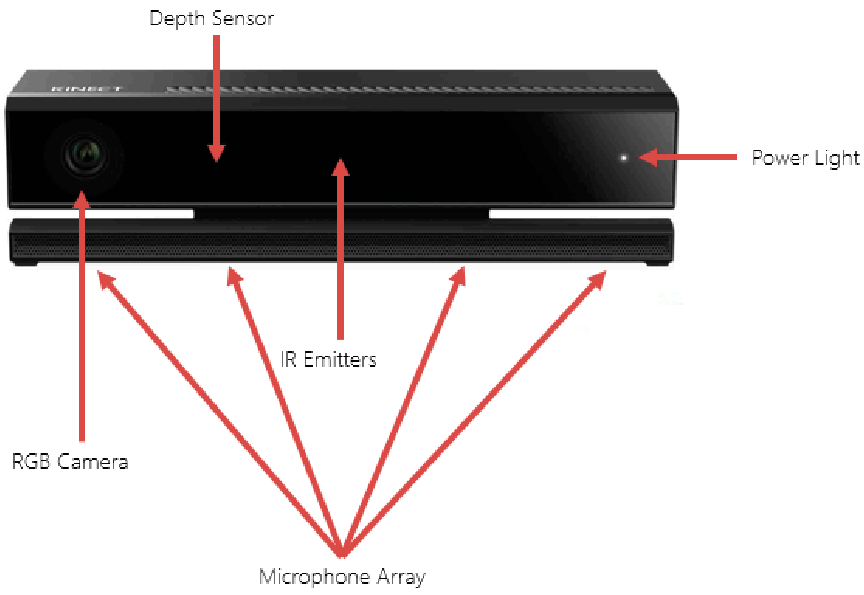
PTZ camera

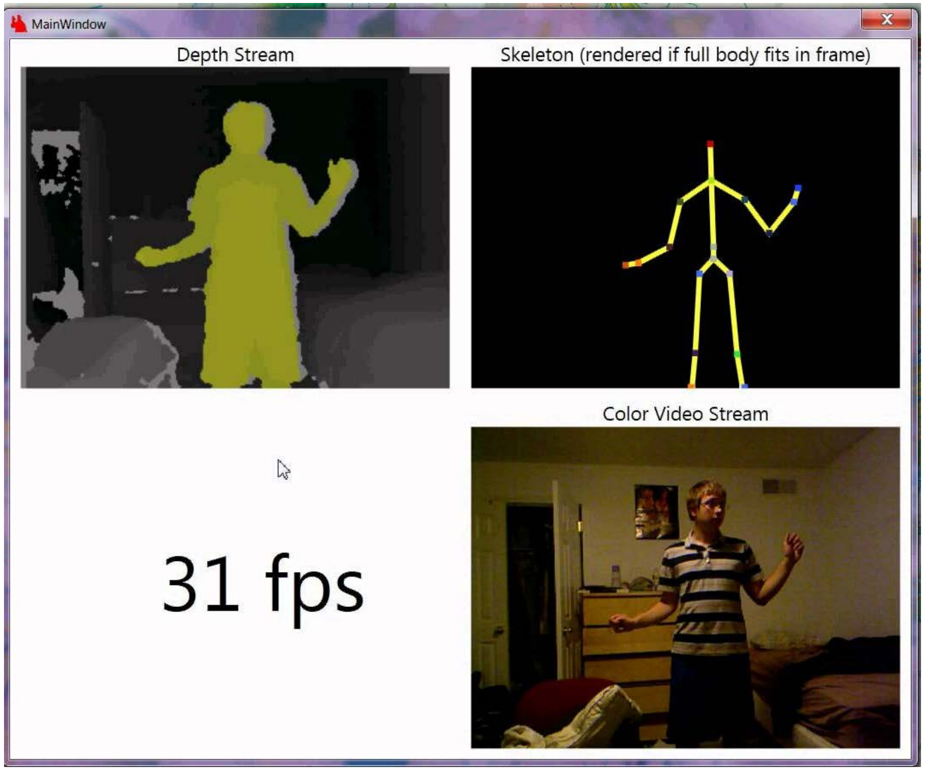
- Matlab code to acquire images/video:

```
% connect to the video stream
cam = ipcam('http://158.42.149.40/mjpg/1/video.mjpg','alumno','tdivtdiv2021');
preview(cam); % open a video preview window
img = snapshot(cam); % get a frame
```

CAMERAS

Kinect v2 device:



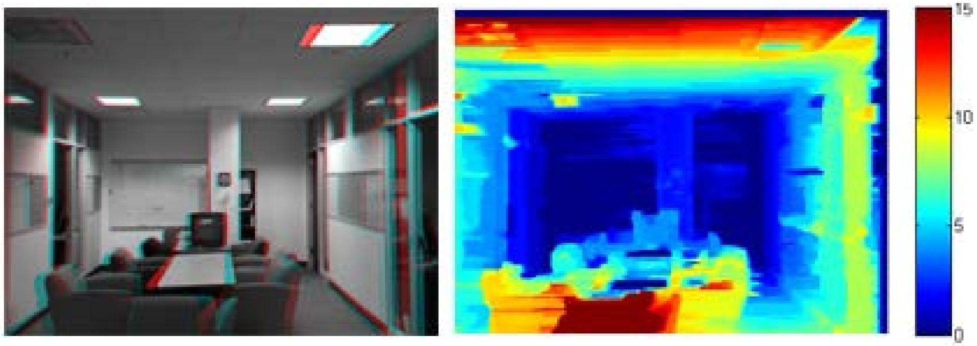


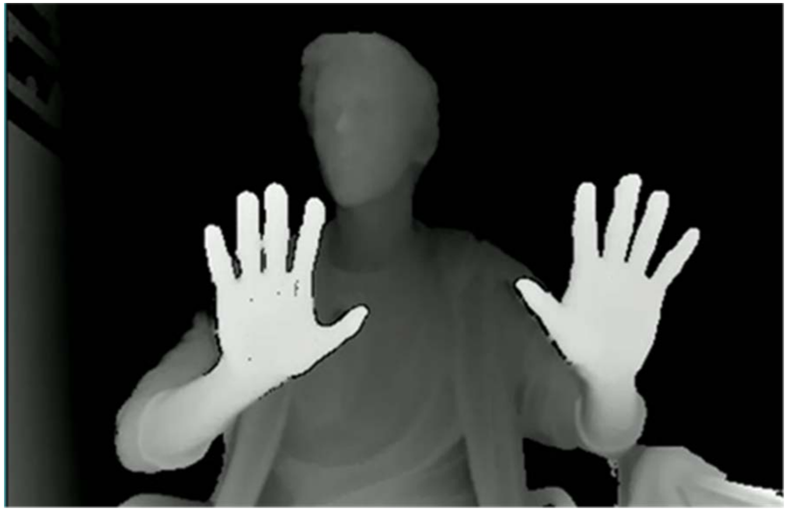
31 fps

- RGB camera: 1920x1080, 30fps
- Depth camera: 512x424, 30fps
- Detection of 25 skeleton joints
- Six people tracking

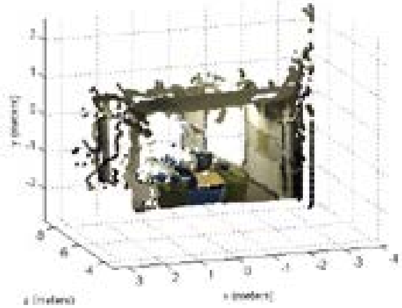
CAMERAS

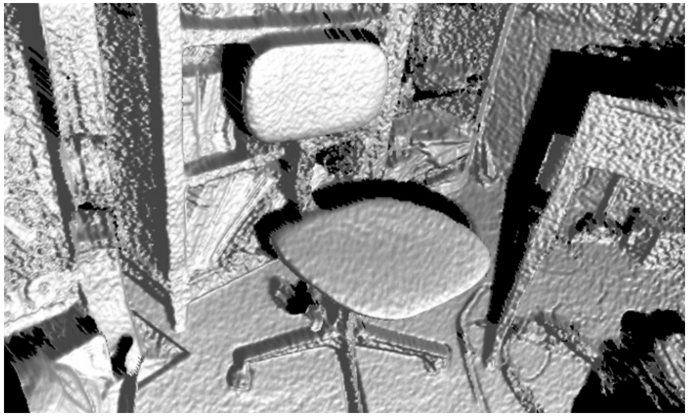
Kinect v2 device:





Depth image





CAMERAS

Kinect v2 device:

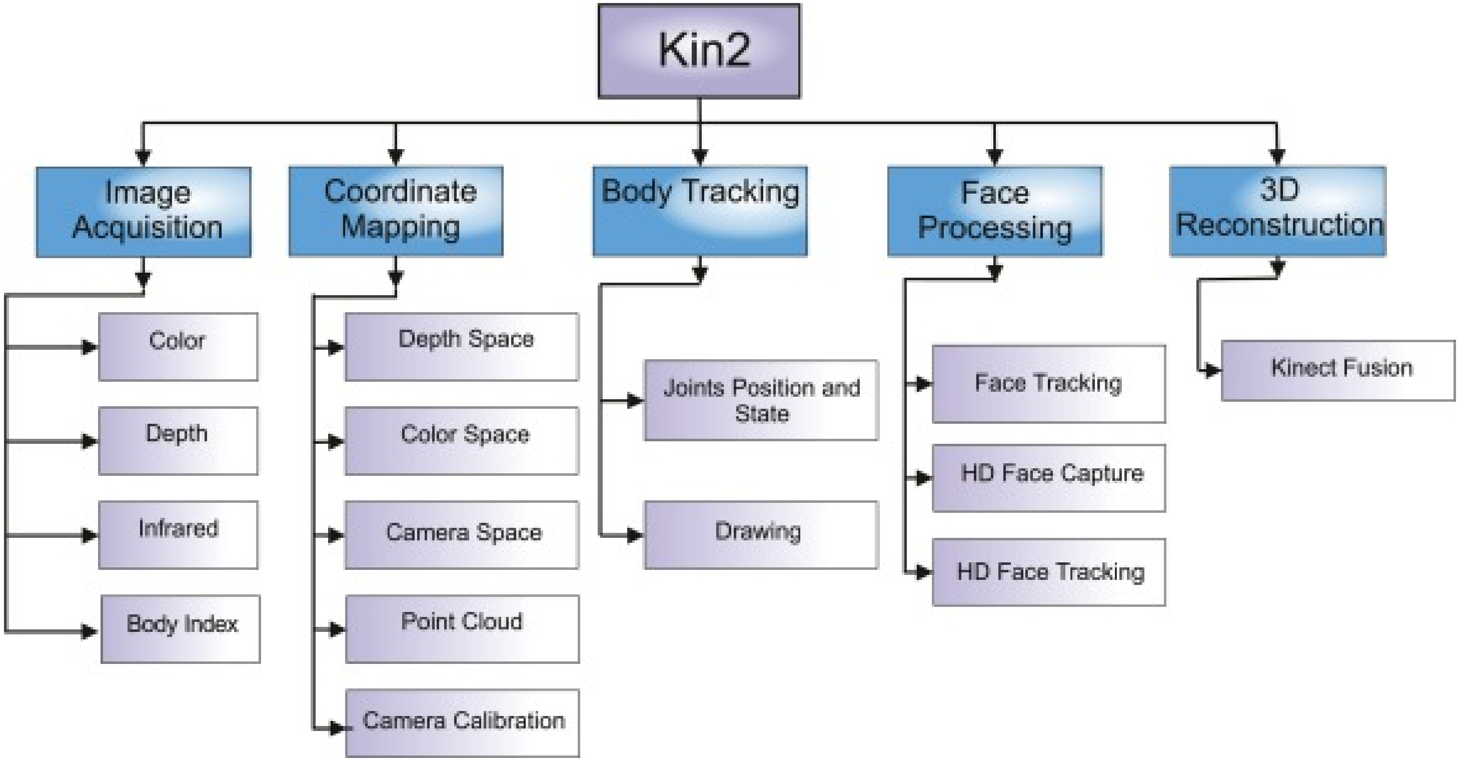
- Install *Image Acquisition Toolbox Support Package for Kinect For Windows Sensor*
- Matlab code to acquire images/video:

```
depthVid = videoinput('kinect', 2); % RGB camera
colorvid = videoinput('kinect', 1); % depth camera
[depthMap, depthMetaData] = getsnapshot(depthVid); % get RGB frame
[image, imageMetaData] = getsnapshot(colorvid); % get depth frame
cuerpo = depthMetaData.DepthJointIndices; % person skeleton
```

CAMERAS

Kinect v2 device:

- Kin2 Toolbox: https://github.com/jrterven/Kin2



CAMERAS

Kinect v2 device:

- Kin2 Toolbox: https://github.com/jrterven/Kin2

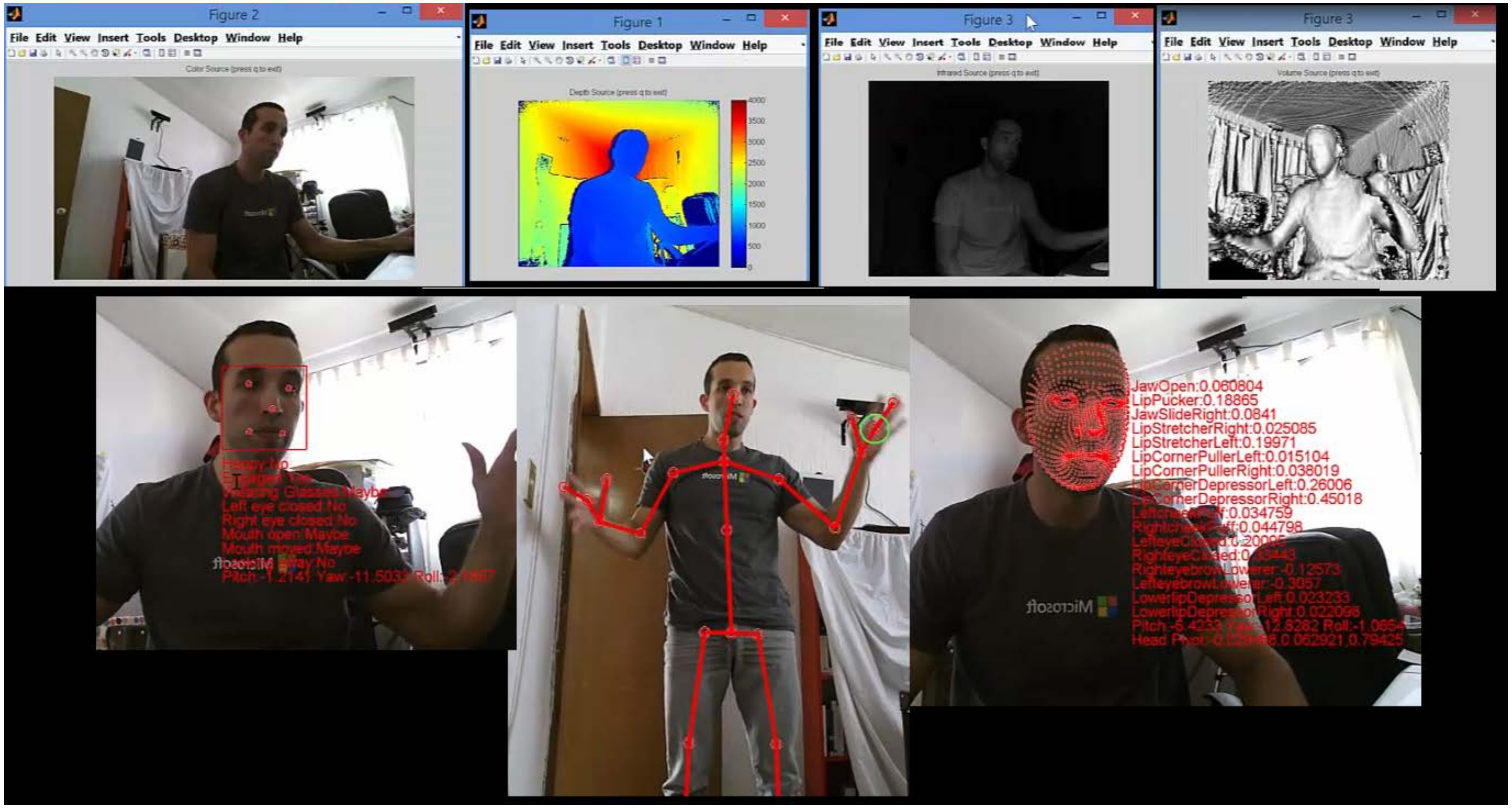


Figure 2
File Edit View Insert Tools Desktop Window Help
Color Source (press q to exit)
Figure 1
File Edit View Insert Tools Desktop Window Help
Depth Source (press q to exit)
4000
3500
3000
2500
2000
1500
1000
500
0
Figure 3
File Edit View Insert Tools Desktop Window Help
Infrared Source (press q to exit)
Figure 3
File Edit View Insert Tools Desktop Window Help
Volume Source (press q to exit)

```
Happy:No
Engaged:Yes
Wearing Glasses:Maybe
Left eye closed:No
Right eye closed:No
Mouth open:Maybe
Mouth moved:Maybe
Looking away:No
Pitch:-1.2141 Yaw:-11.5033 Roll:-2.1567
```

```
JawOpen:0.060804
LipPucker:0.18865
JawSlideRight:0.0841
LipStretcherRight:0.025085
LipStretcherLeft:0.19971
LipCornerPullerLeft:0.015104
LipCornerPullerRight:0.038019
LipCornerDepressorLeft:0.26006
LipCornerDepressorRight:0.45018
LeftcheekPuff:0.034759
RightcheekPuff:0.044798
LefteyeClosed:0.20095
RighteyeClosed:0.33443
RighteyebrowLowerer:-0.12573
LefteyebrowLowerer:-0.3057
LowerlipDepressorLeft:0.023233
LowerlipDepressorRight:0.022098
Pitch:-6.4233 Yaw:-12.6282 Roll:-1.0654
Head Pivot:-0.029468,0.062921,0.79425
```

CAMERAS

Basler industrial cameras:

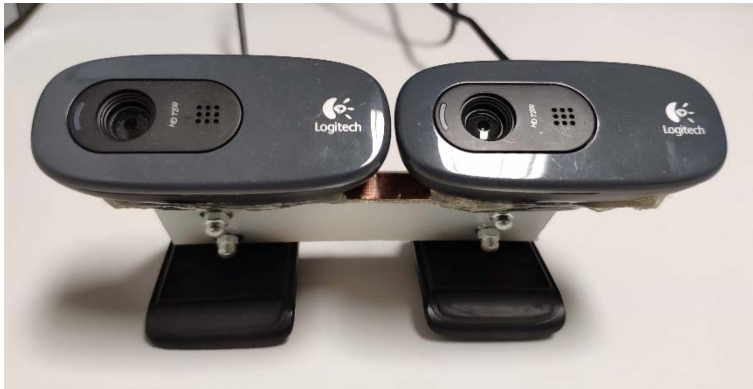
- Monochrome camera
- 3840x2748 px, 10fps
- GigE Vision Interface

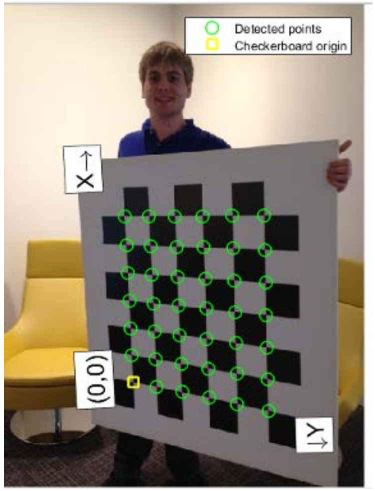
- Install *GigE Vision Support from Image Acquisition Toolbox*
- Matlab code to acquire images/video:

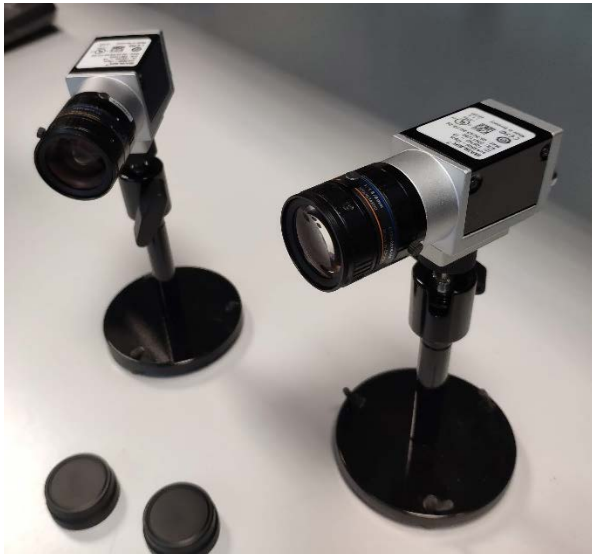
```
gigecamlist; % camera list
cam = gigecam(1); % connect to the camera (first one)
preview(cam); % open a video preview window
img = snapshot(cam); % get a frame
imshow(img); % show the captured frame
g.ScalingHorizontalAbs=0.25; g.ScalingVerticalAbs=0.25; % 1/4 resolution
```

CAMERAS

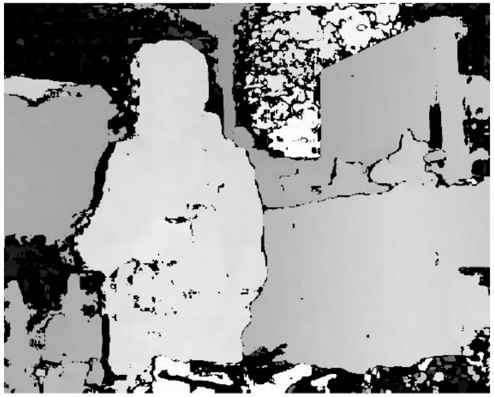
Stereo Vision: 2 cameras & calibration







Depth map



Blur effect