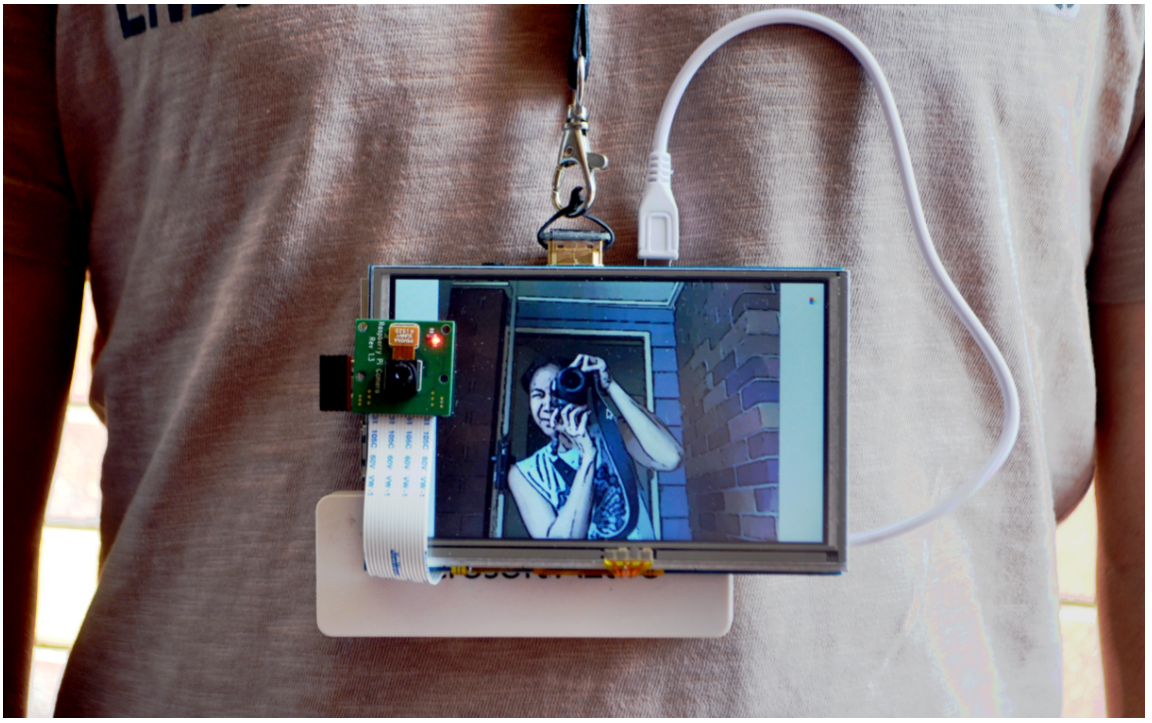
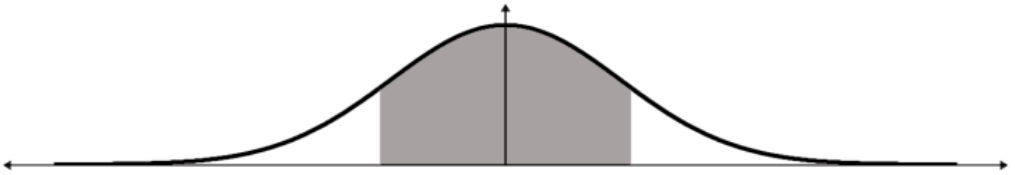
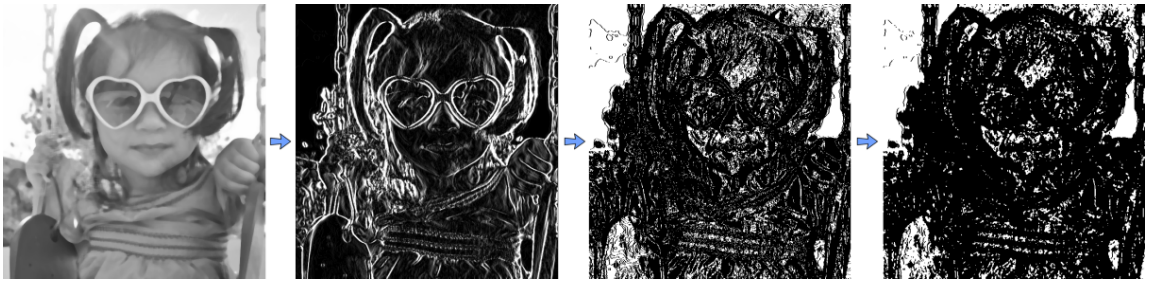
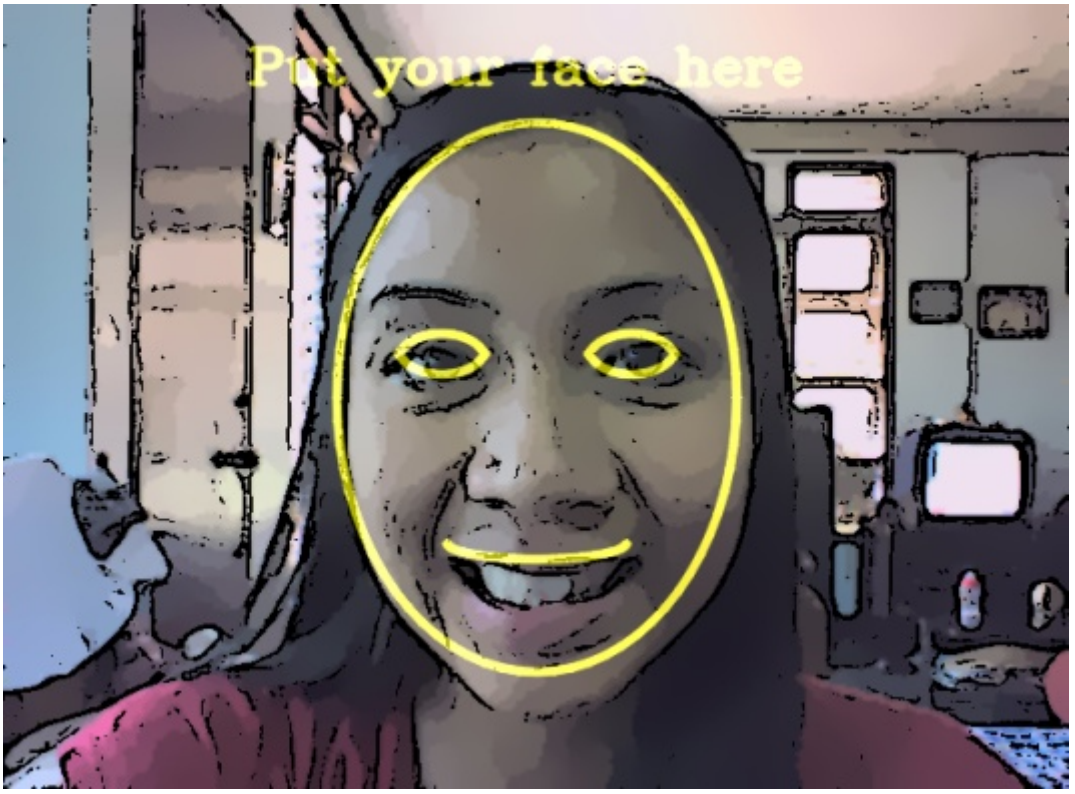
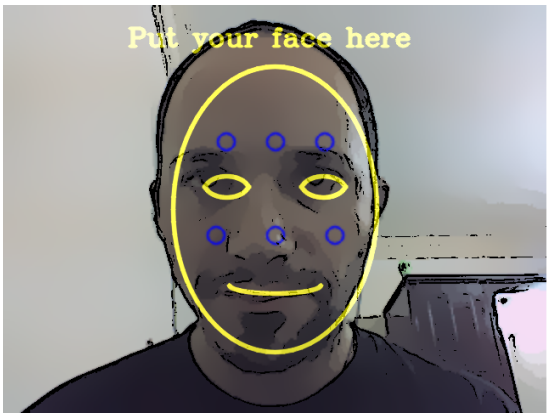
Chapter 1: Cartoonifier and Skin Changer for Raspberry Pi

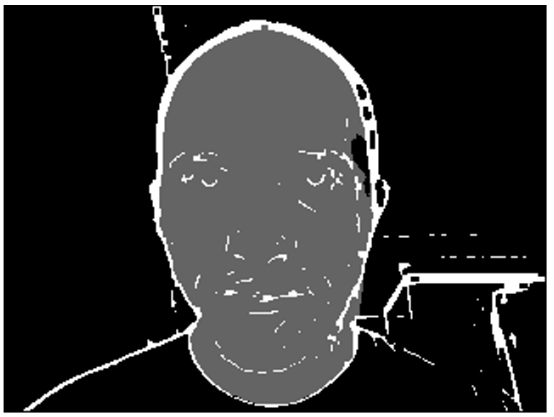


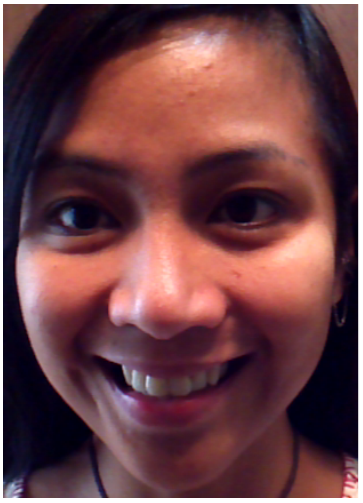


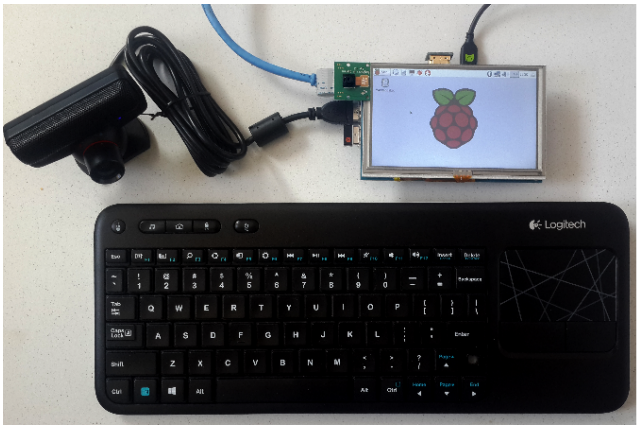


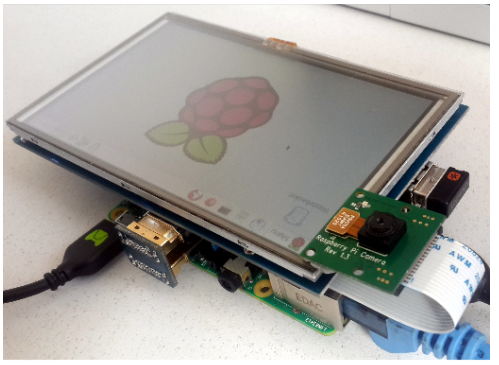

Put your face here

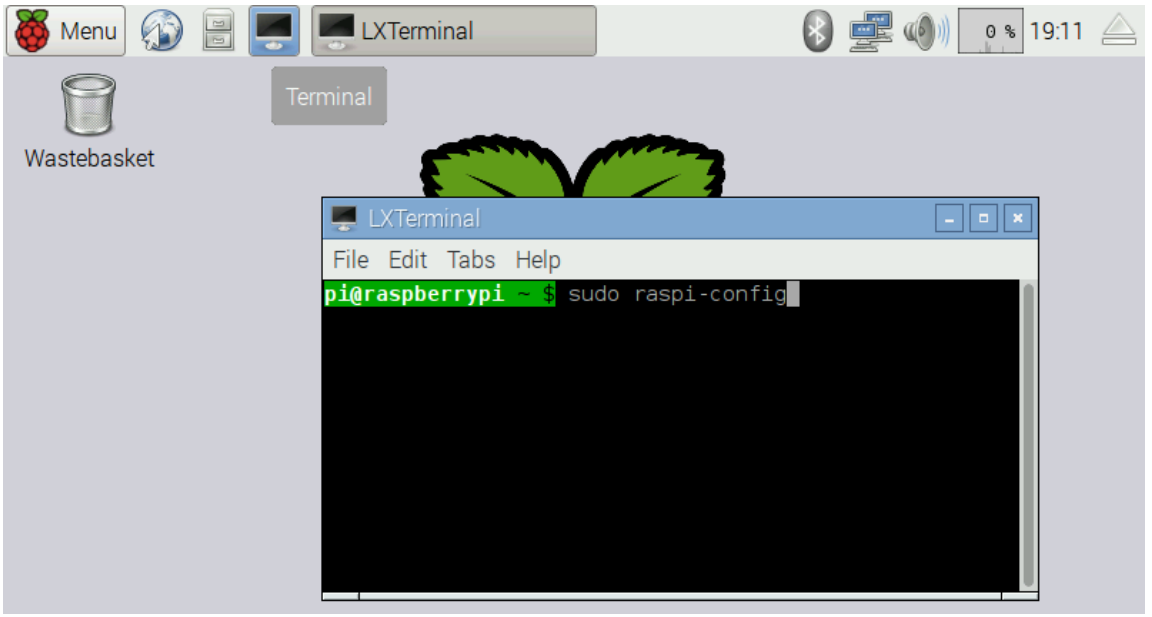

Put your face here

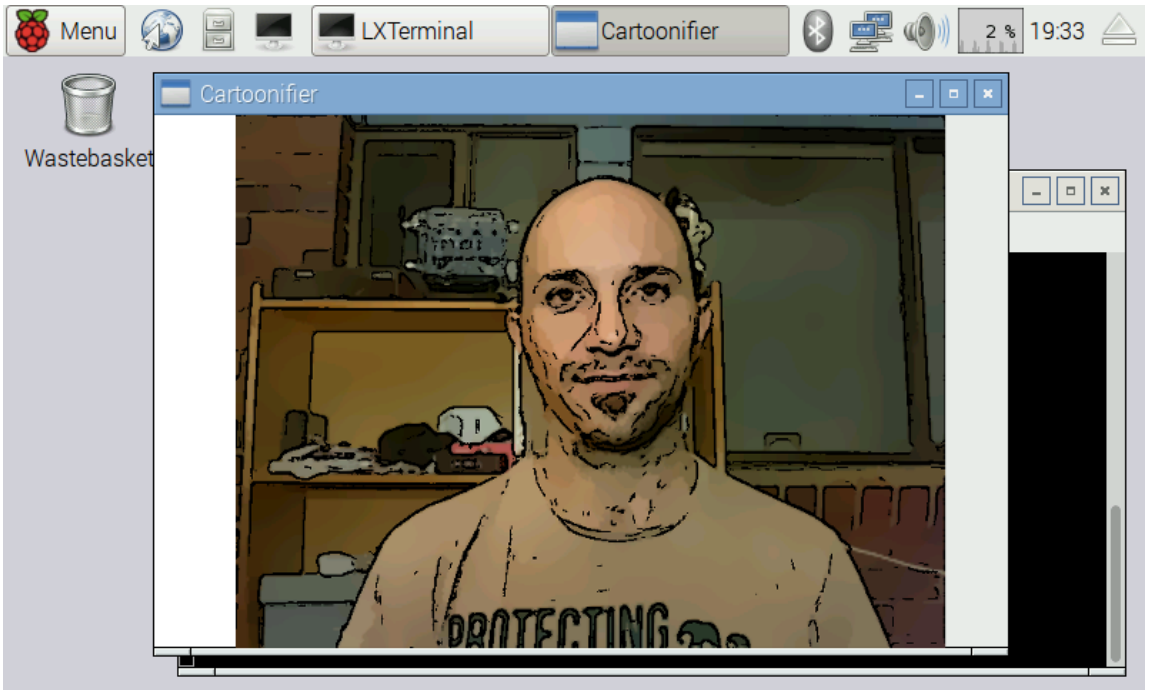


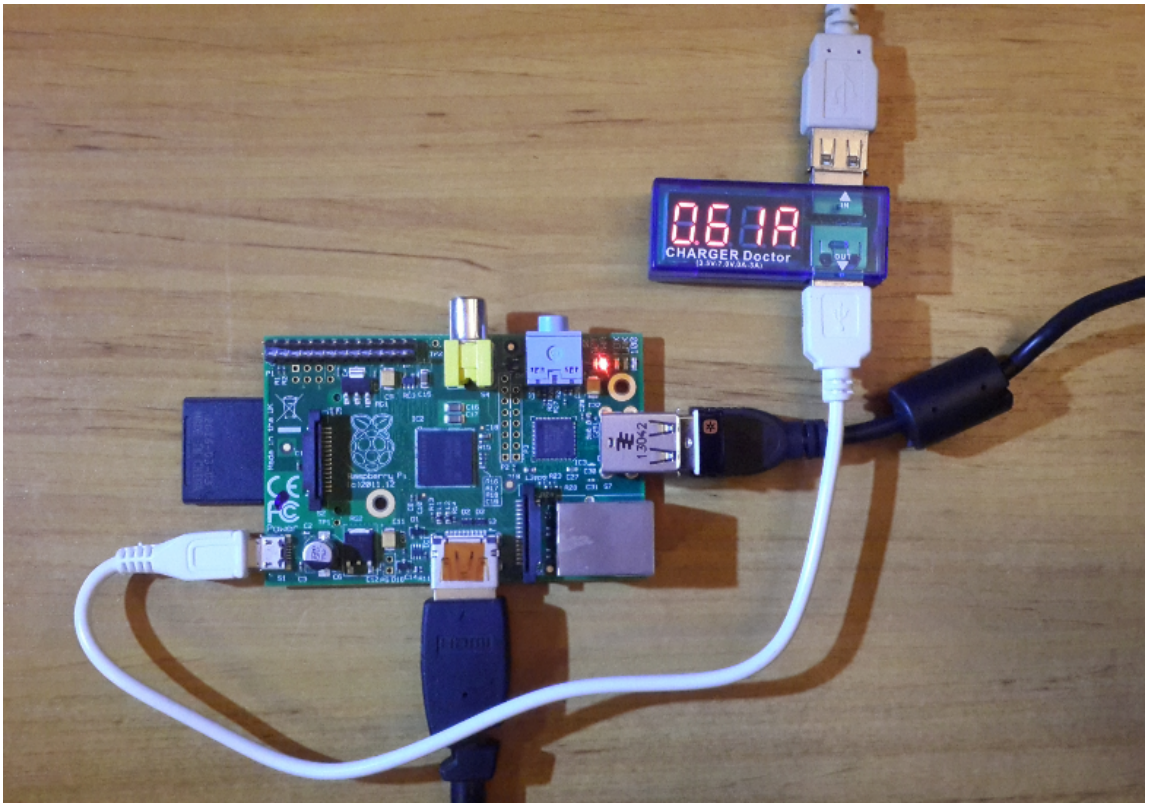



Logitech

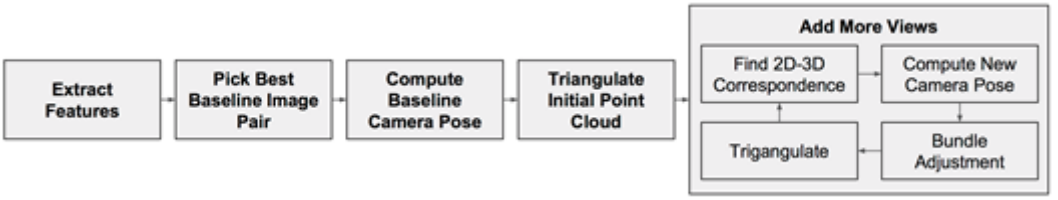


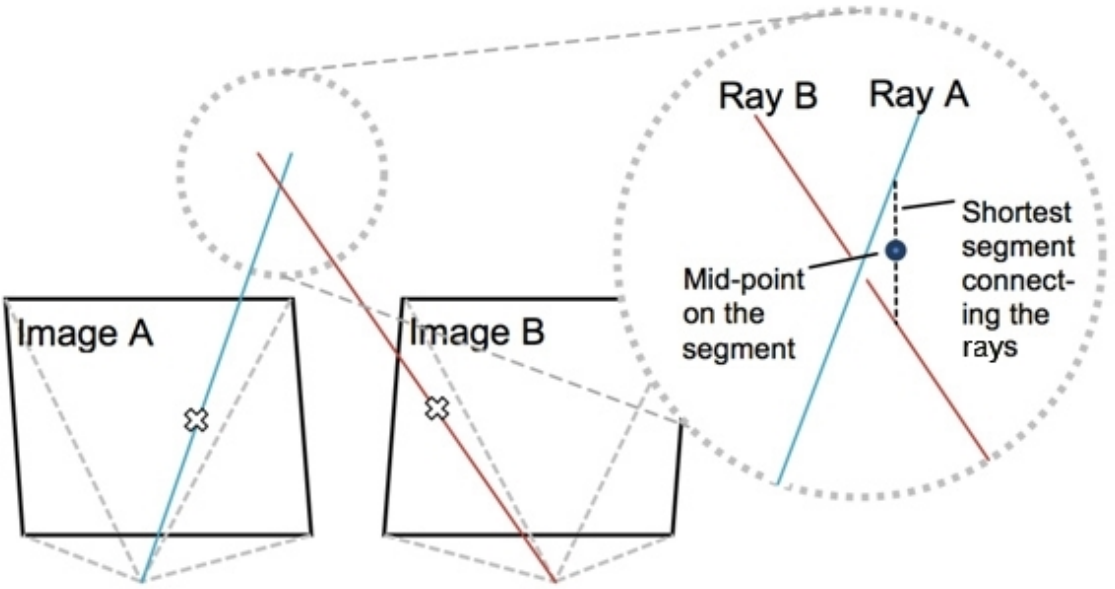

Menu
LXTerminal
0 %
19:11
Terminal
Wastebasket
LXTerminal
File Edit Tabs Help
pi@raspberrypi ~ $ sudo raspi-config

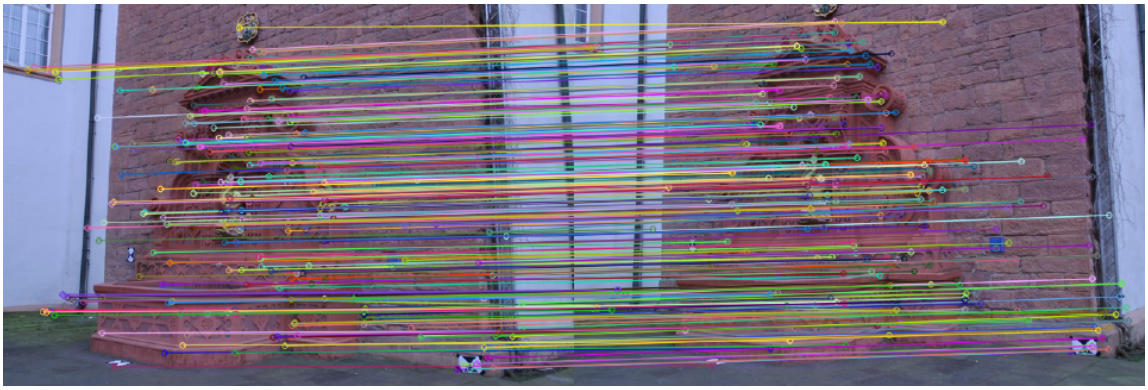

Menu
LXTerminal
Cartoonifier
2 %
19:33
Wastebasket
Cartoonifier

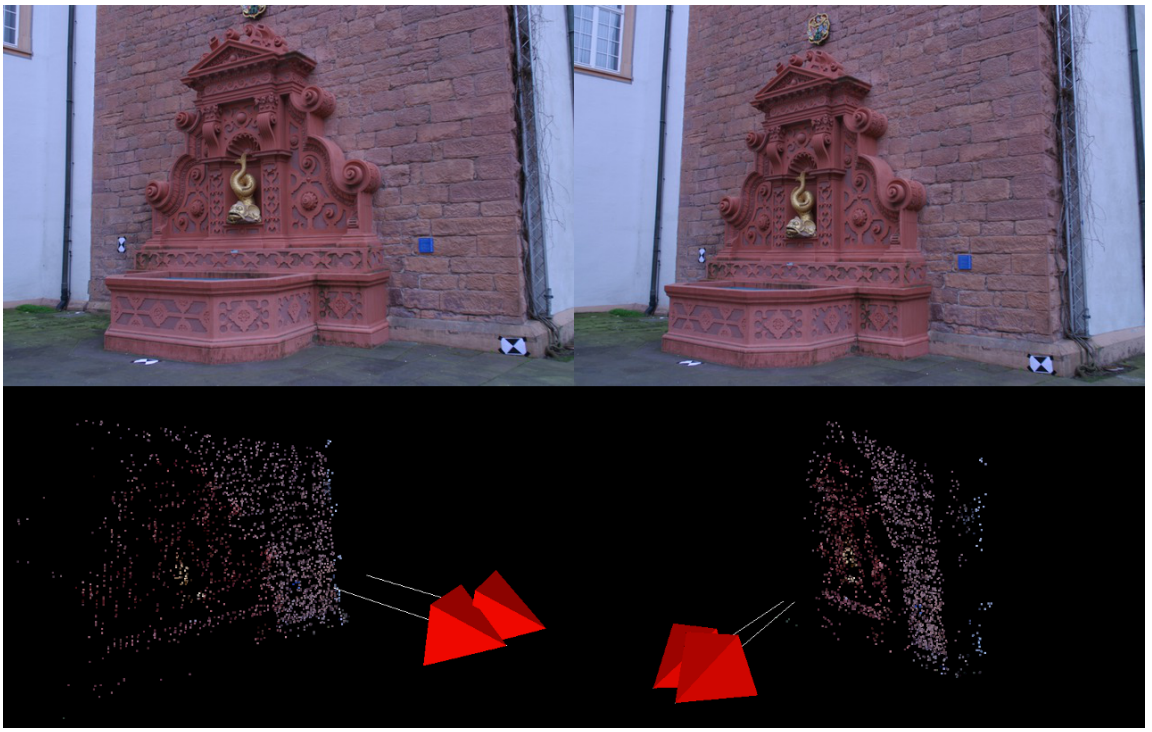

0.61A
CHARGER Doctor

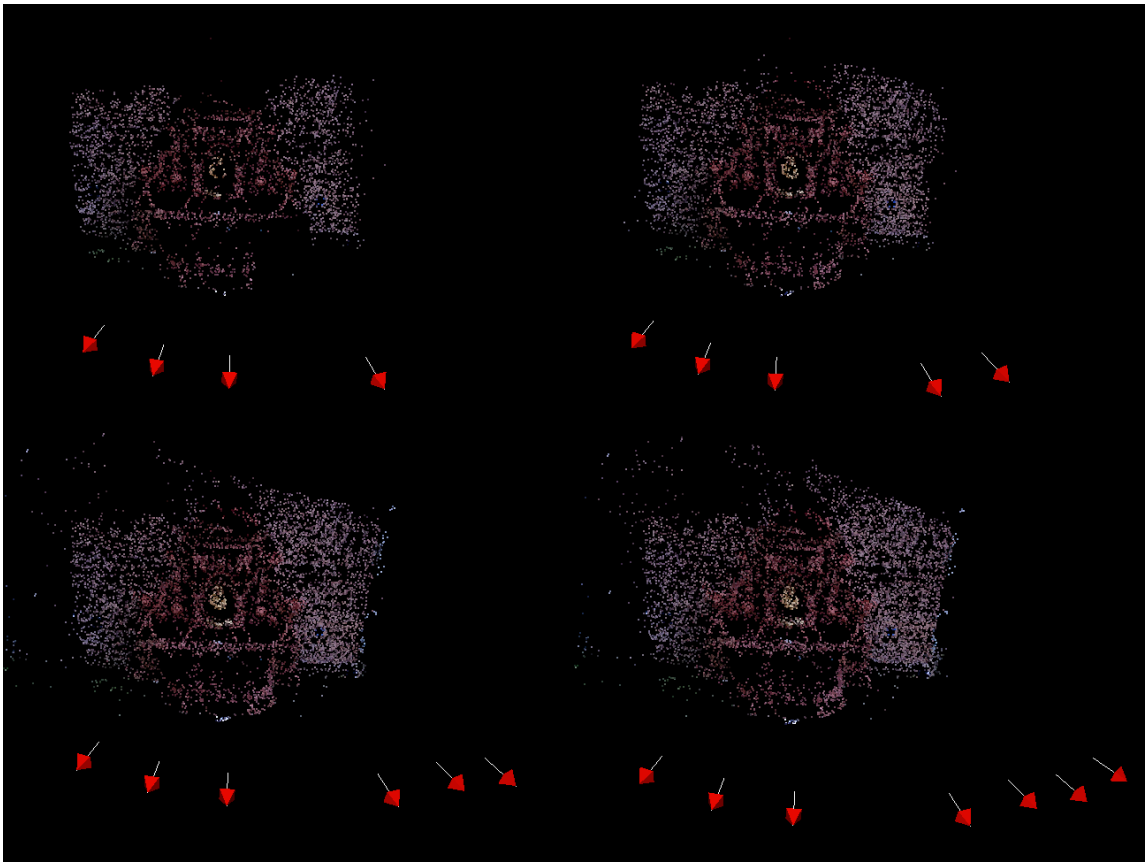
Chapter 2: Exploring Structure from Motion Using OpenCV



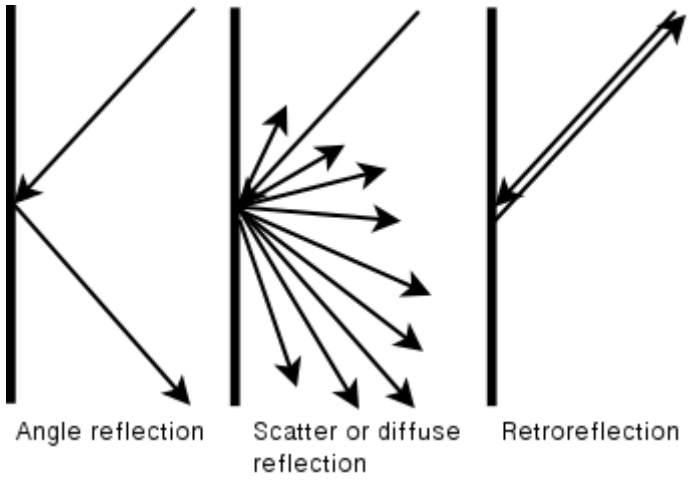


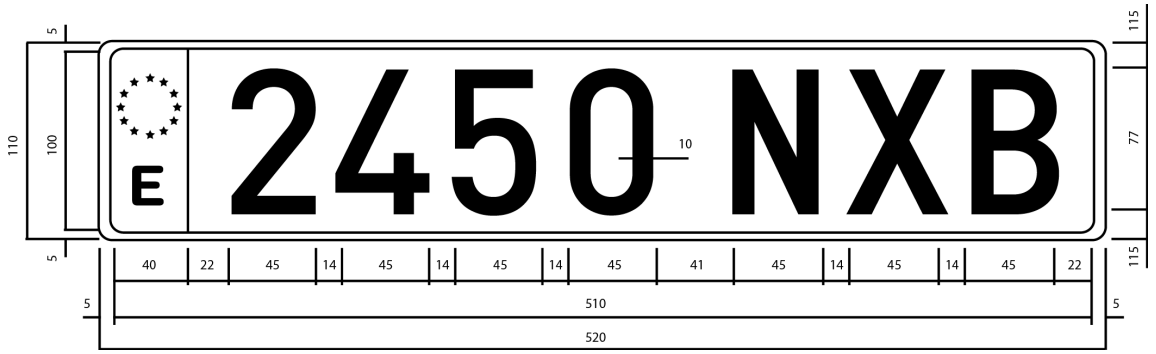


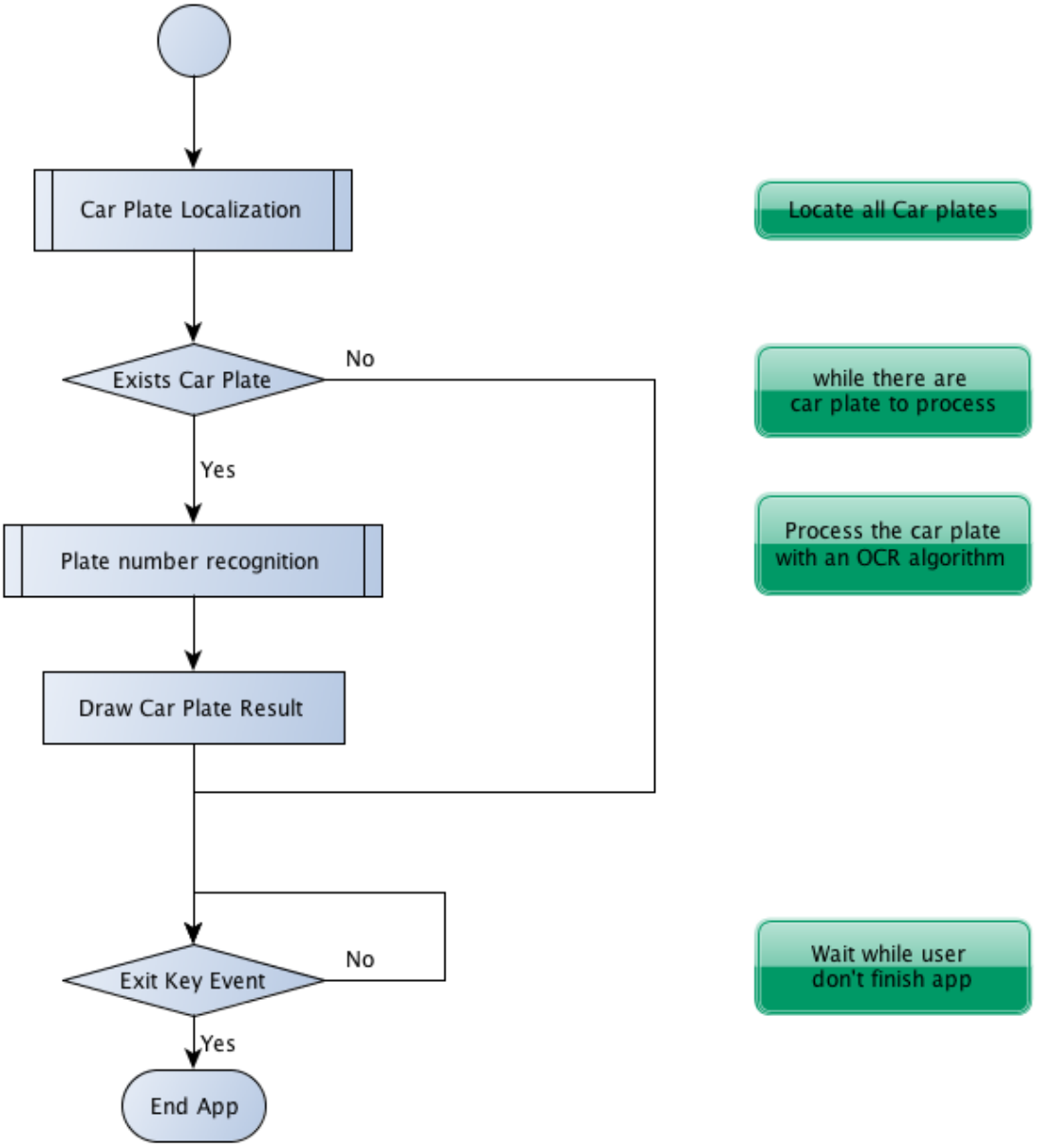


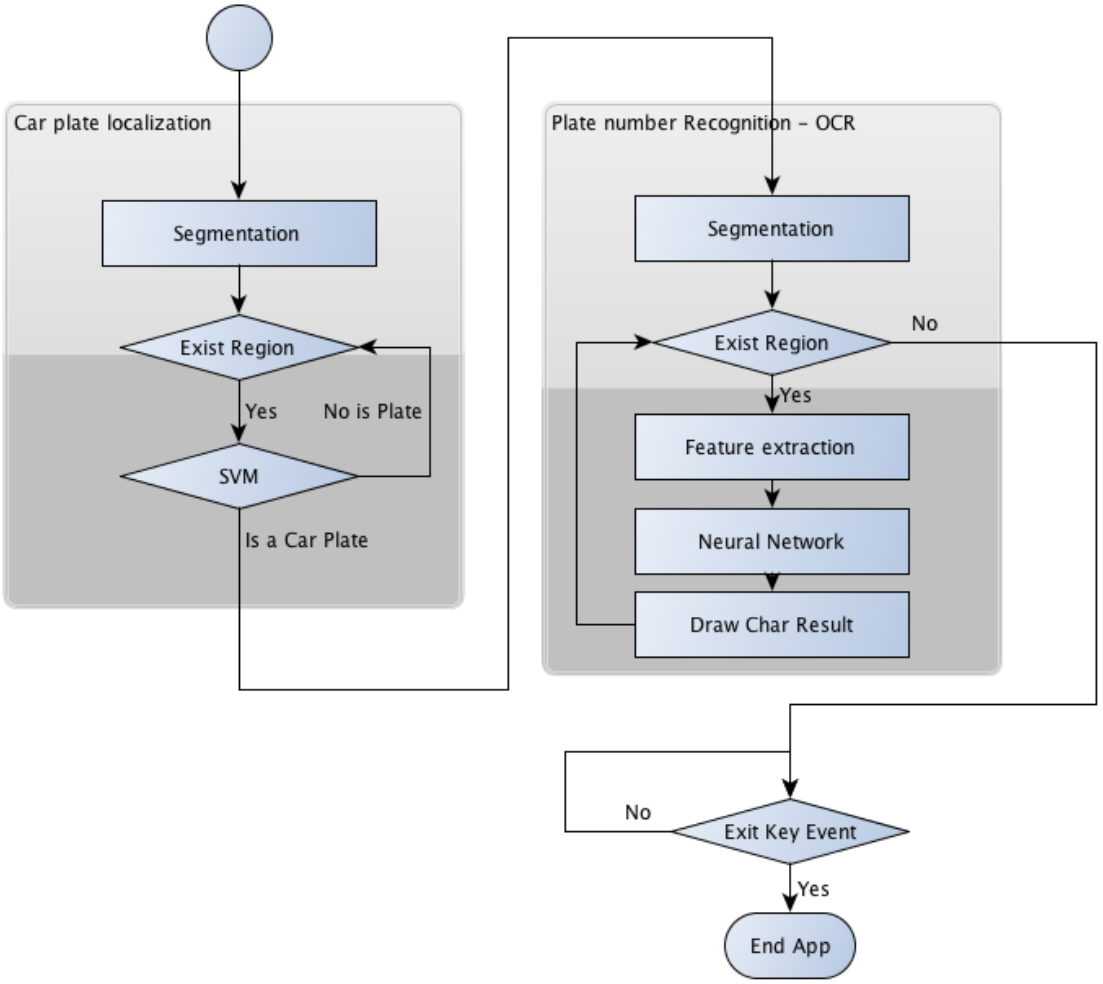


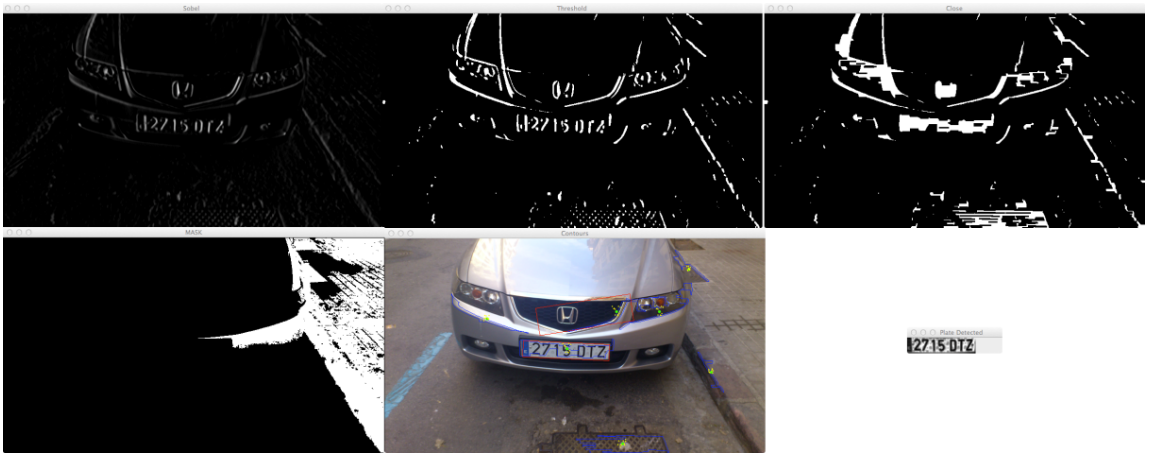
Chapter 3: Number Plate Recognition using SVM and Neural Network

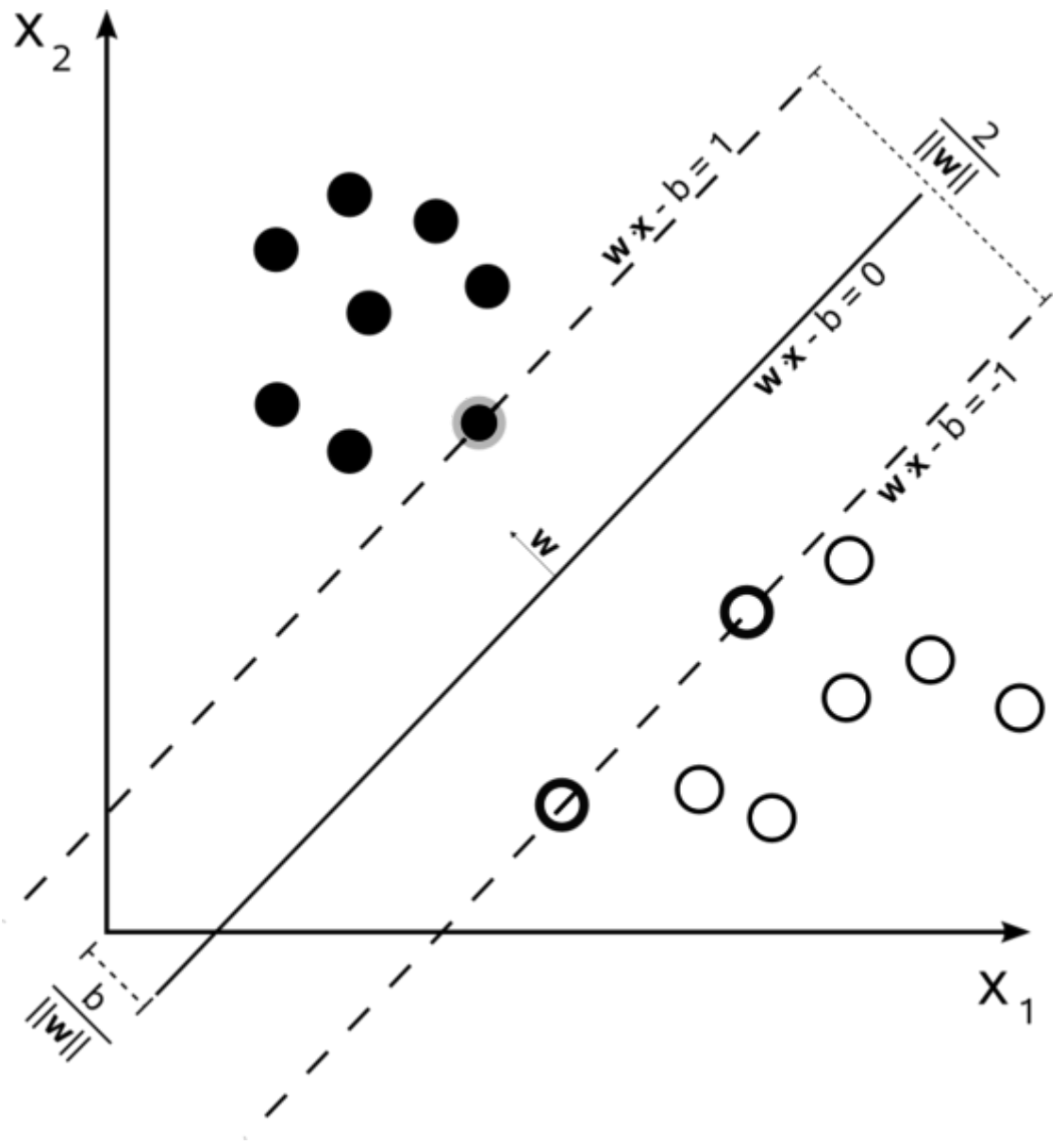



E
2450 NXB
10
5
110
100
5
115
77
115
40
22
45
14
45
14
45
14
45
41
45
14
45
14
45
22
5
510
5
520

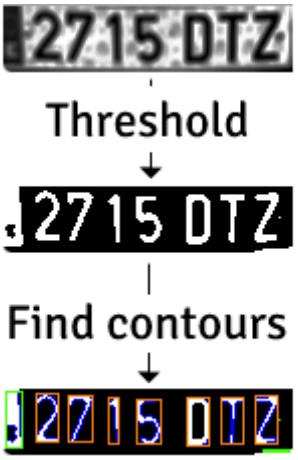

Car Plate Localization
Exists Car Plate
No
Yes
Plate number recognition
Draw Car Plate Result
Exit Key Event
No
Yes
End App
Locate all Car plates
while there are
car plate to process
Process the car plate
with an OCR algorithm
Wait while user
don't finish app

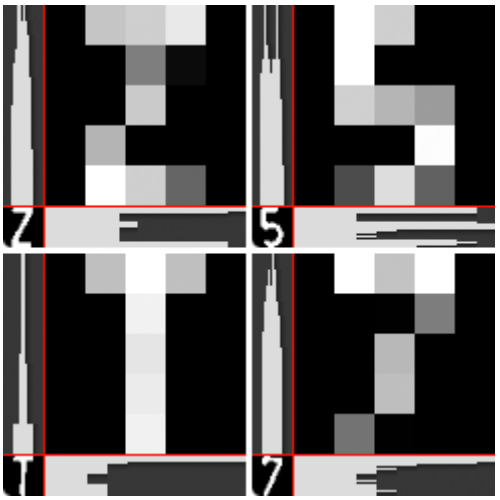

Car plate localization
Segmentation
Exist Region
Yes
No is Plate
SVM
Is a Car Plate
Plate number Recognition - OCR
Segmentation
Exist Region
No
Yes
Feature extraction
Neural Network
Draw Char Result
No
Exit Key Event
Yes
End App

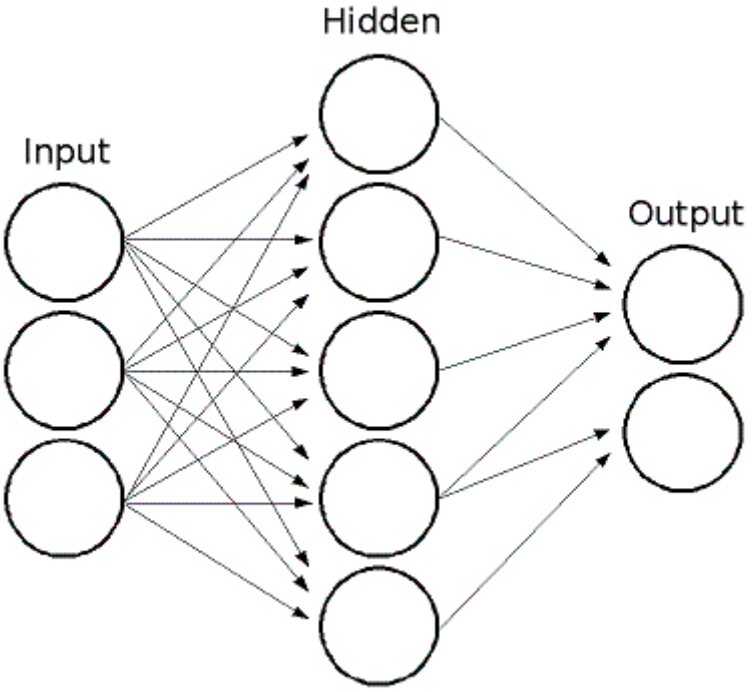

Sobel
Threshold
Close
MASK
Contours
2715-DTZ
Plate Detected
2715-DTZ

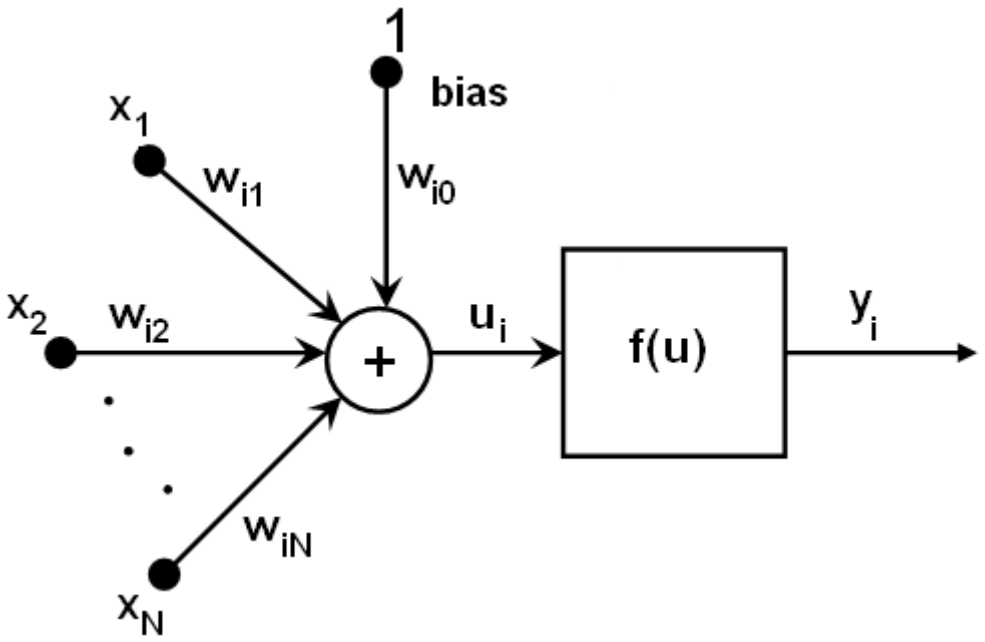

X2
X1
w x - b = 1
w x - b = 0
w x - b = -1
2 / ||w||
w
b / ||w||

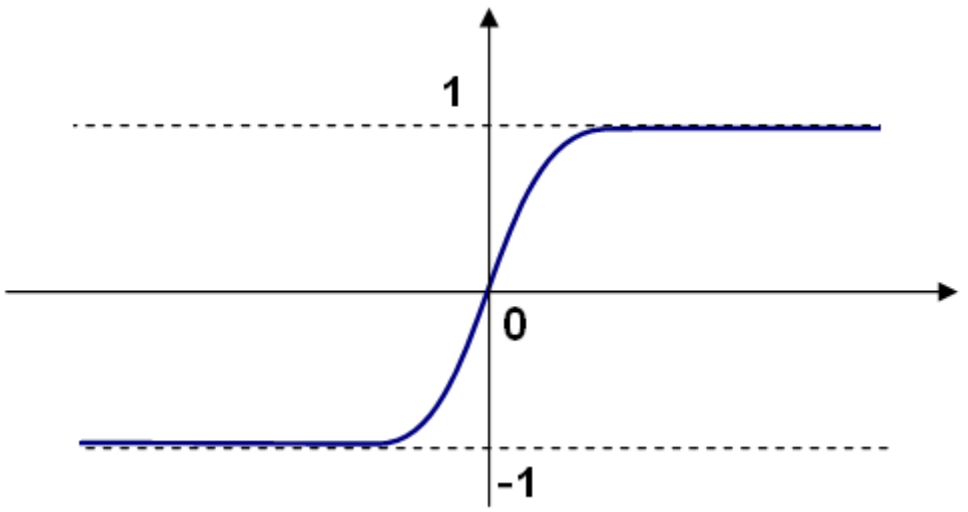

1148 CJS
6275 FND
7713 DSY
8660 GLY
9362 BLM
0260 FXX
2900 GLB
6989 GKV
7878 BXF
8895 DNJ
9588 DWV
0688 GVH
3028 BYS
7021 DSK
8066 GWL
8995 CCN
8235 BVJ
2488 DBC
0489 GDW
7215 BGN
8085 FNC
9057 BHS
7215 BGN
2715 DTZ
3839 CWG
7384 HGY
8308 FLP
9169 BWH
0022 FRR
3099 BYV


2715 DTZ
Threshold
2715 DTZ
Find contours

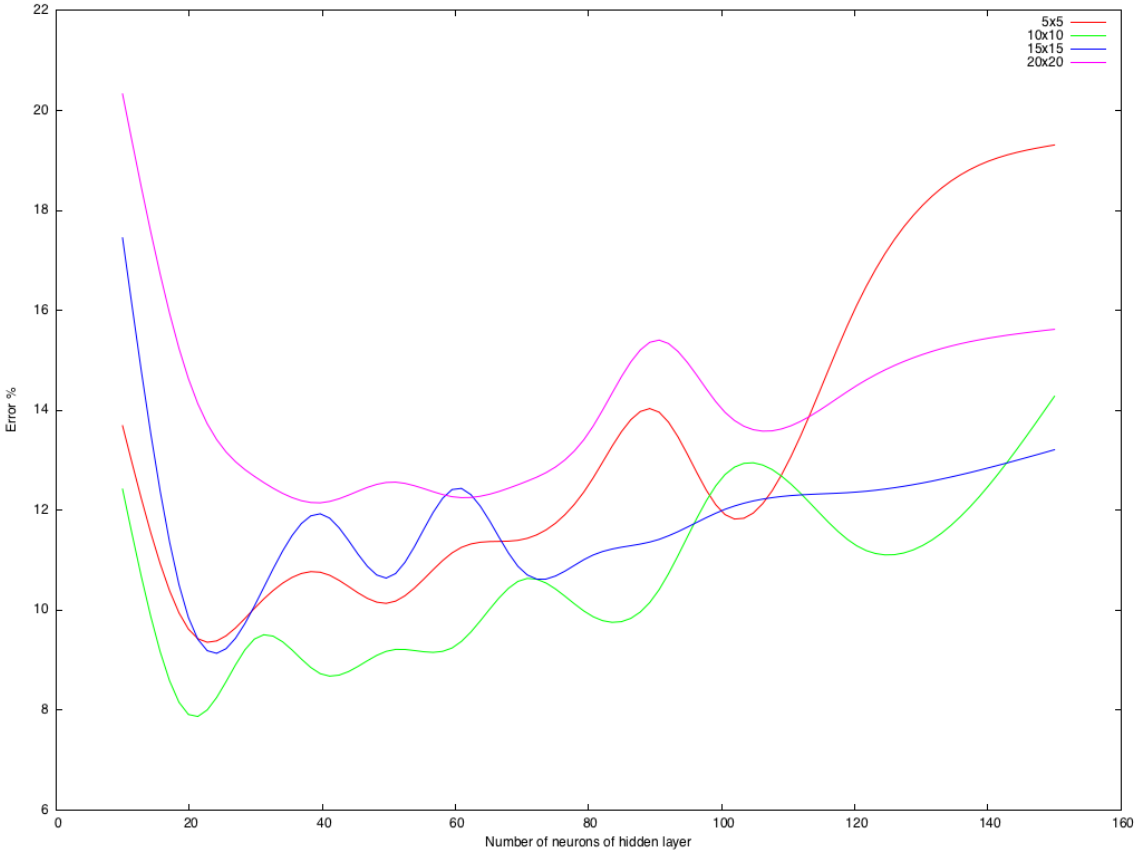



Hidden
Input
Output

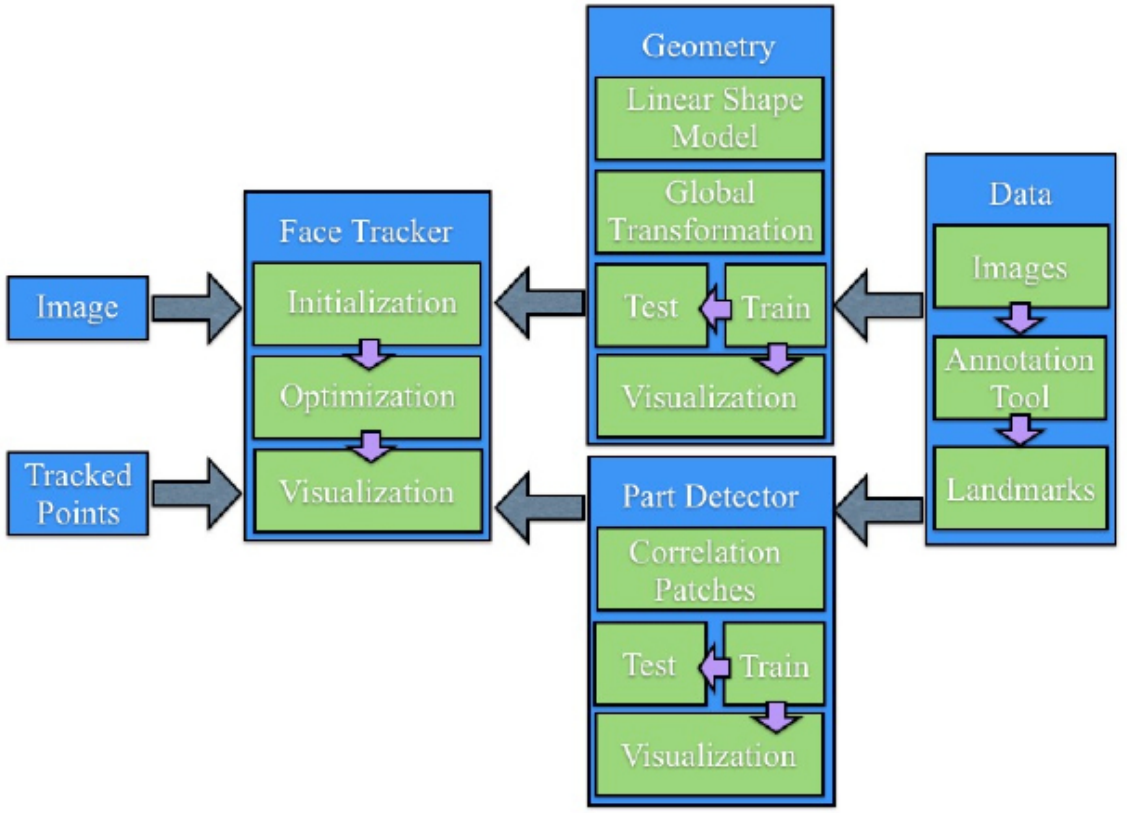

1
bias
x1
wi1
wi0
x2
wi2
+
ui
f(u)
yi
wiN
xN

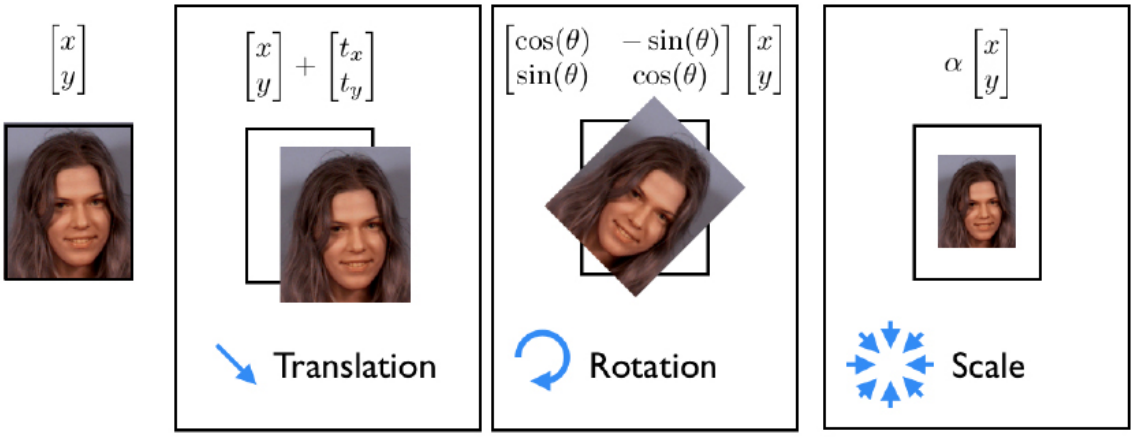

1
0
-1

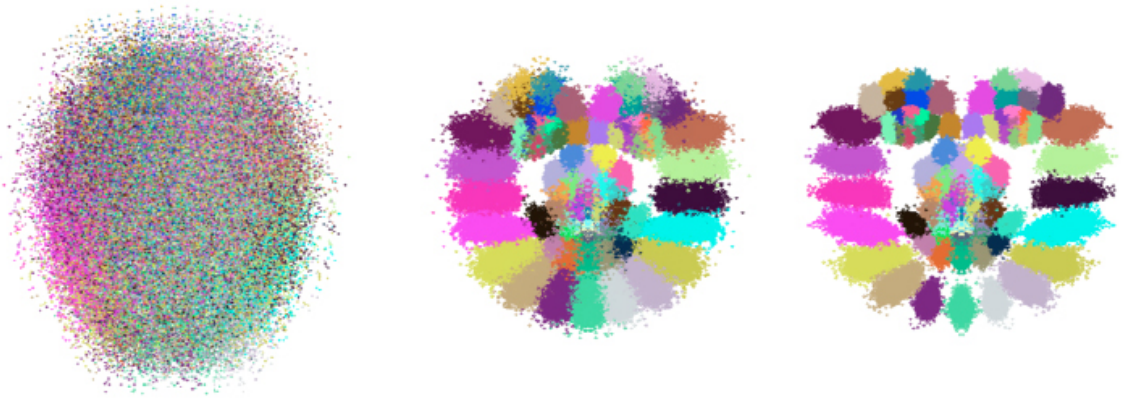

2715DTZ
2715 DTZ


5x5
10x10
15x15
20x20
Error %
Number of neurons of hidden layer
22
20
18
16
14
12
10
8
6
0
20
40
60
80
100
120
140
160

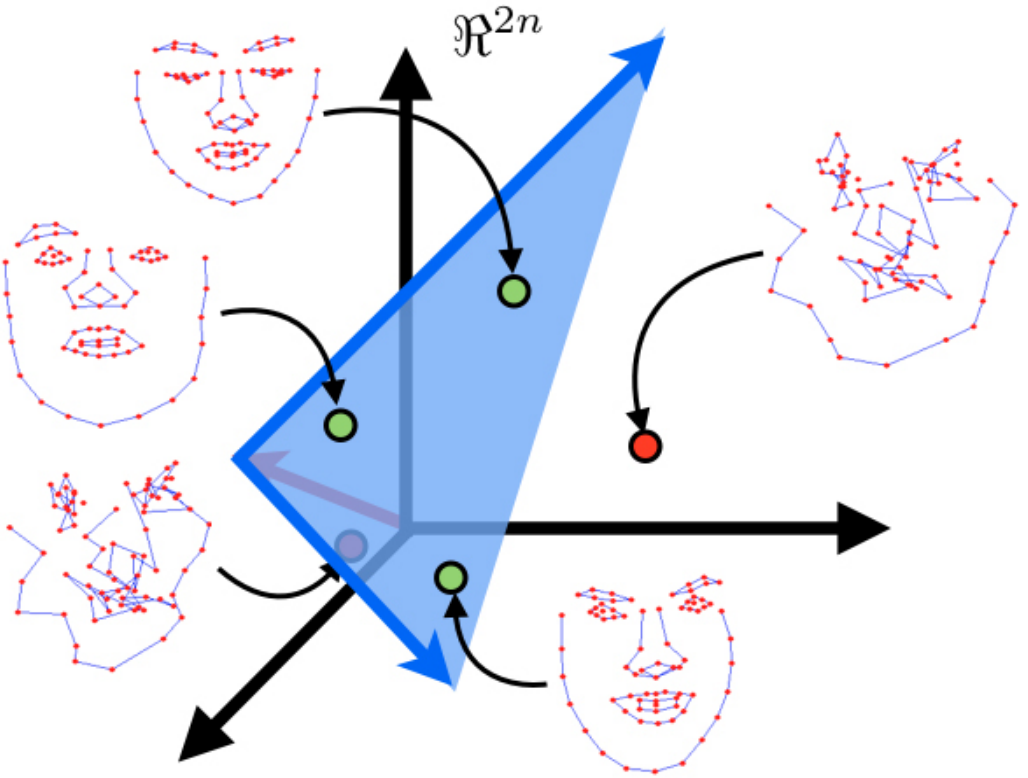
Chapter 4: Non-Rigid Face Tracking

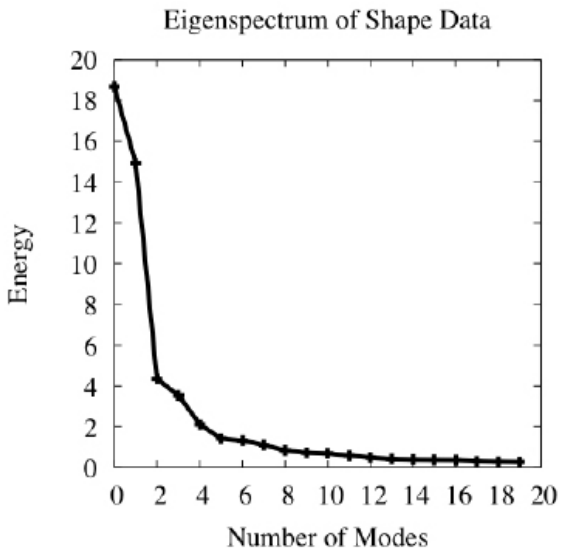


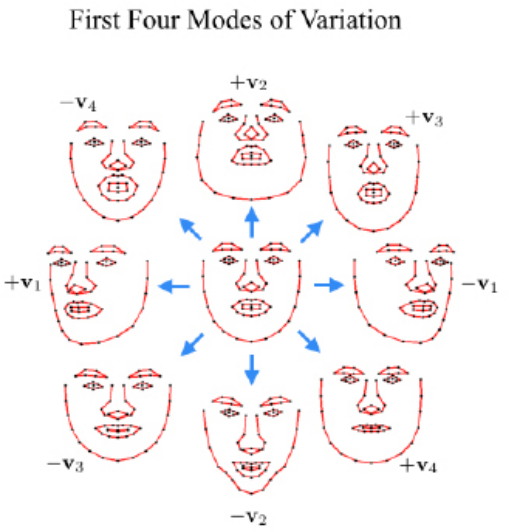




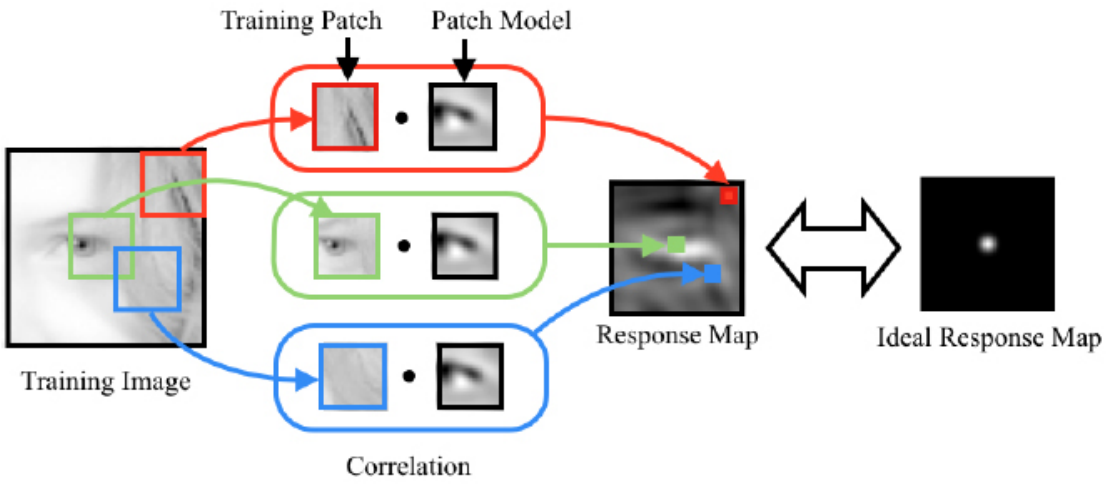
Unaligned Shapes | Translation Aligned | Procrustes Aligned


$\Re^{2n}$

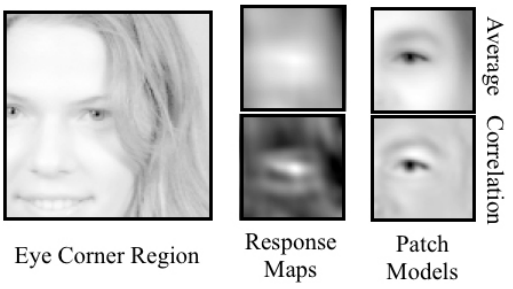

Eigenspectrum of Shape Data
Energy
Number of Modes
0
2
4
6
8
10
12
14
16
18
20


First Four Modes of Variation
$+\mathbf{v}_2$
$-\mathbf{v}_4$
$+\mathbf{v}_3$
$+\mathbf{v}_1$
$-\mathbf{v}_1$
$-\mathbf{v}_3$
$+\mathbf{v}_4$
$-\mathbf{v}_2$


0
4σ
8σ
12σ
16σ


Training Patch
Patch Model
Training Image
Response Map
Ideal Response Map
Correlation

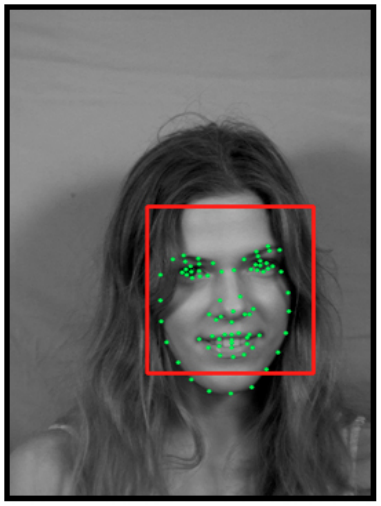

Raw
Log Scale


Eye Corner Region
Response Maps
Patch Models
Average Correlation
Chin Region
Response Maps
Patch Models
Average Correlation

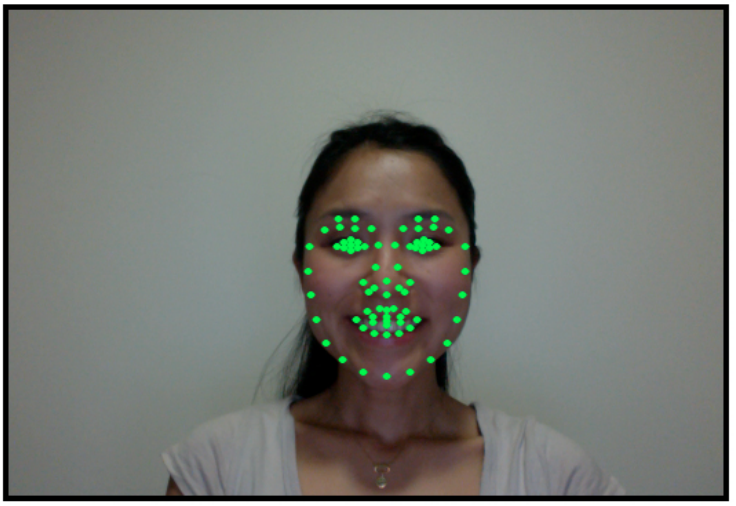
(41x41)

(21x21)

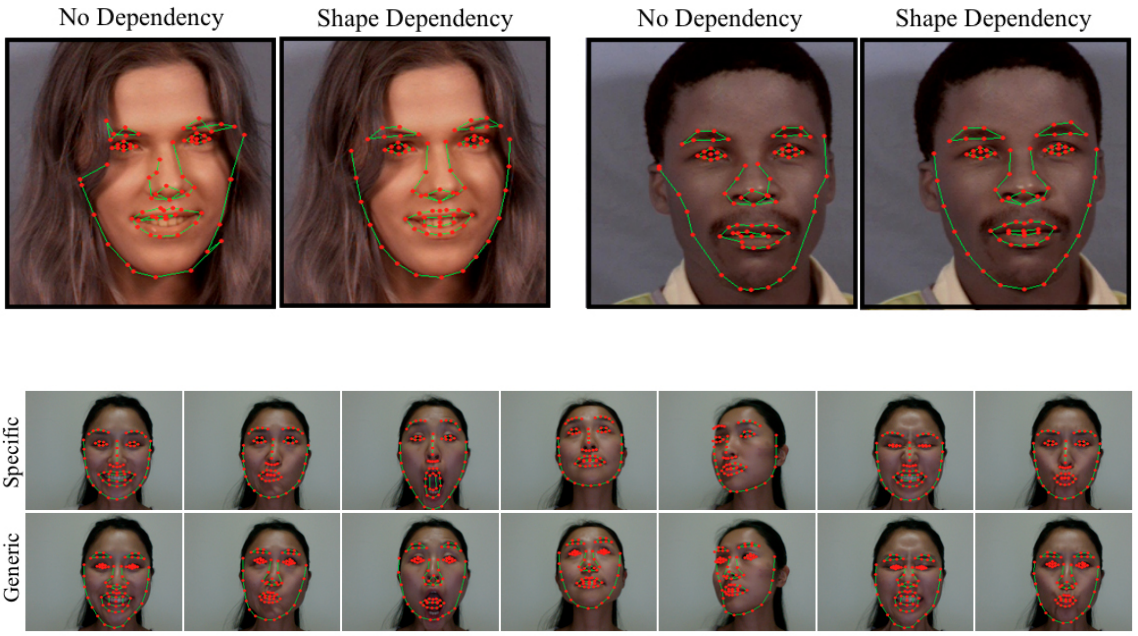
(11x11)



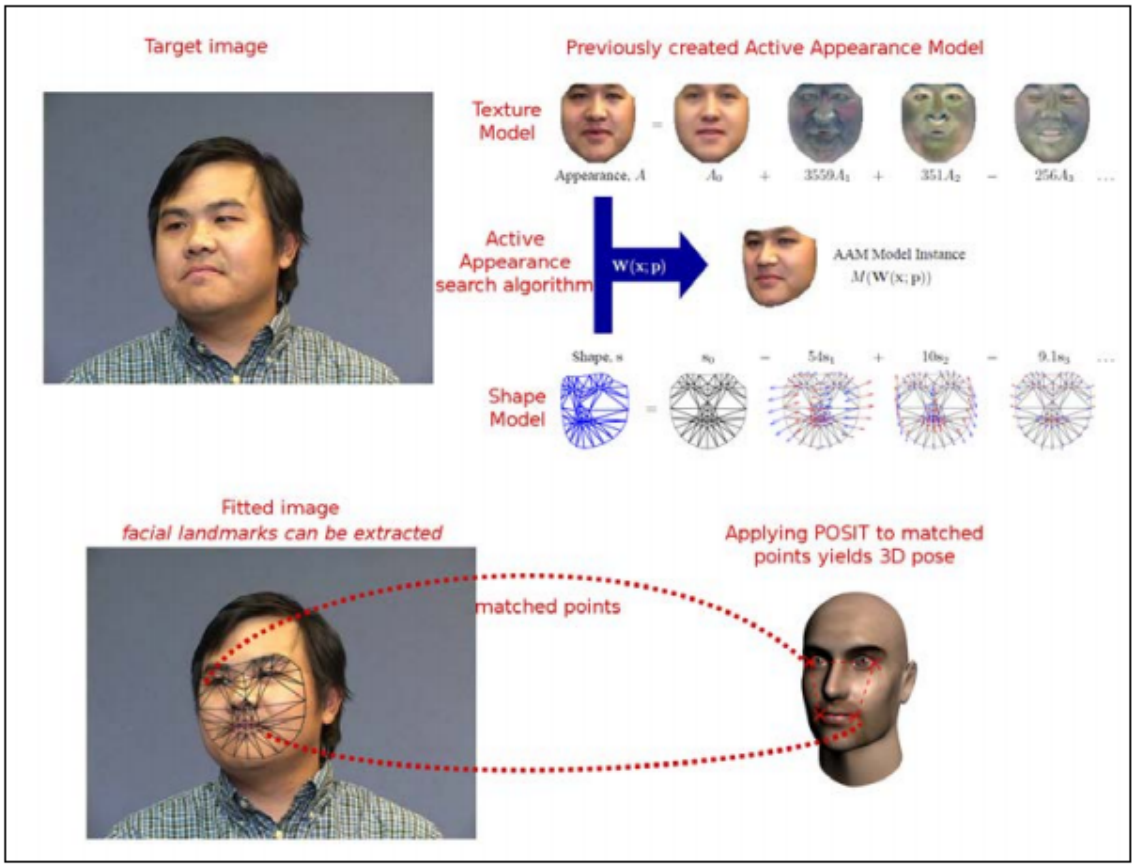
Training Image

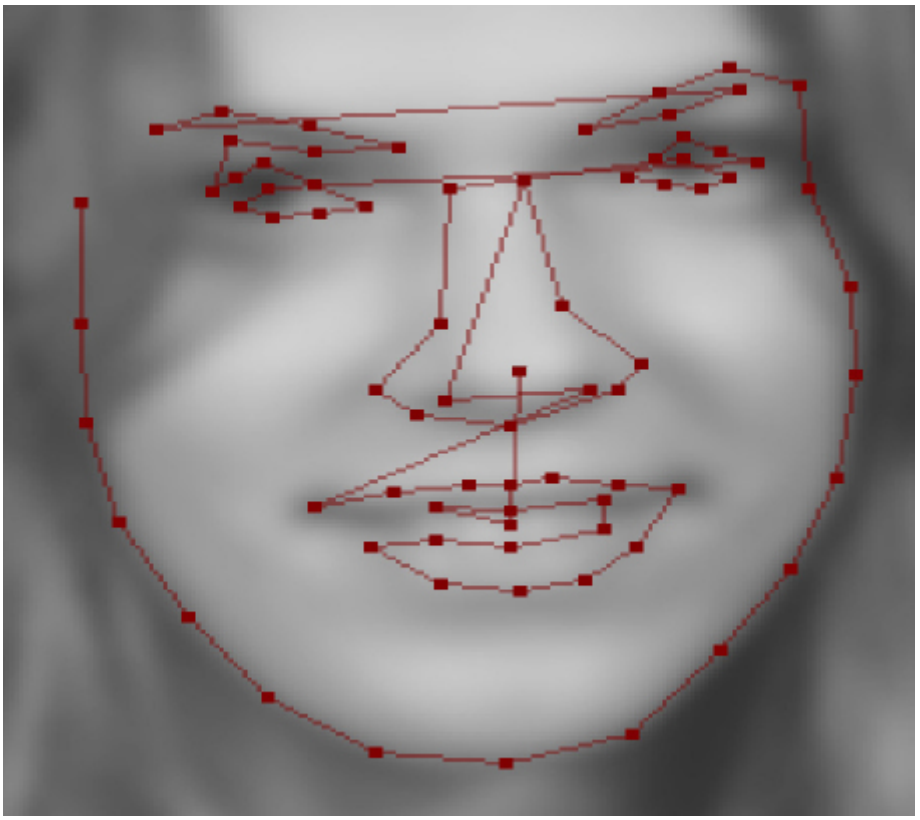


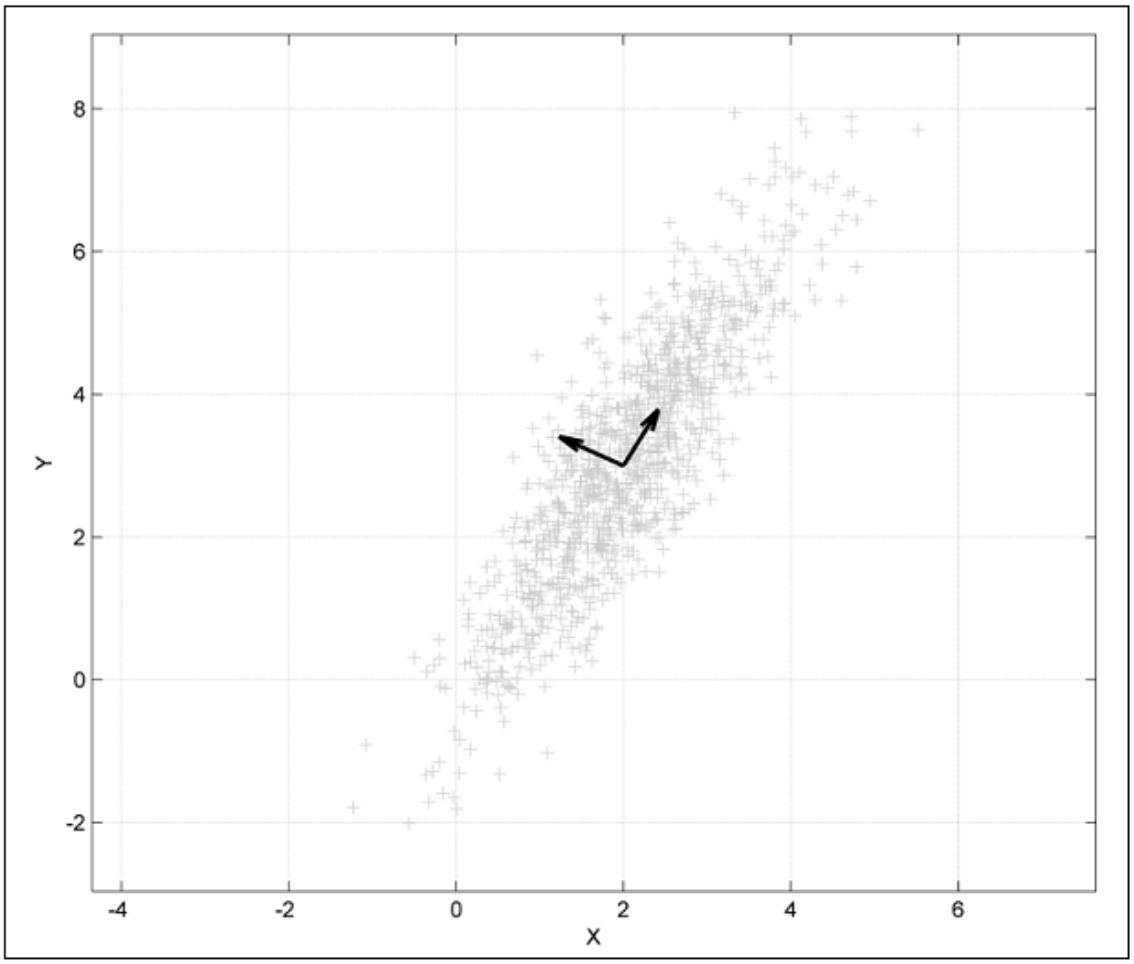
Test Image

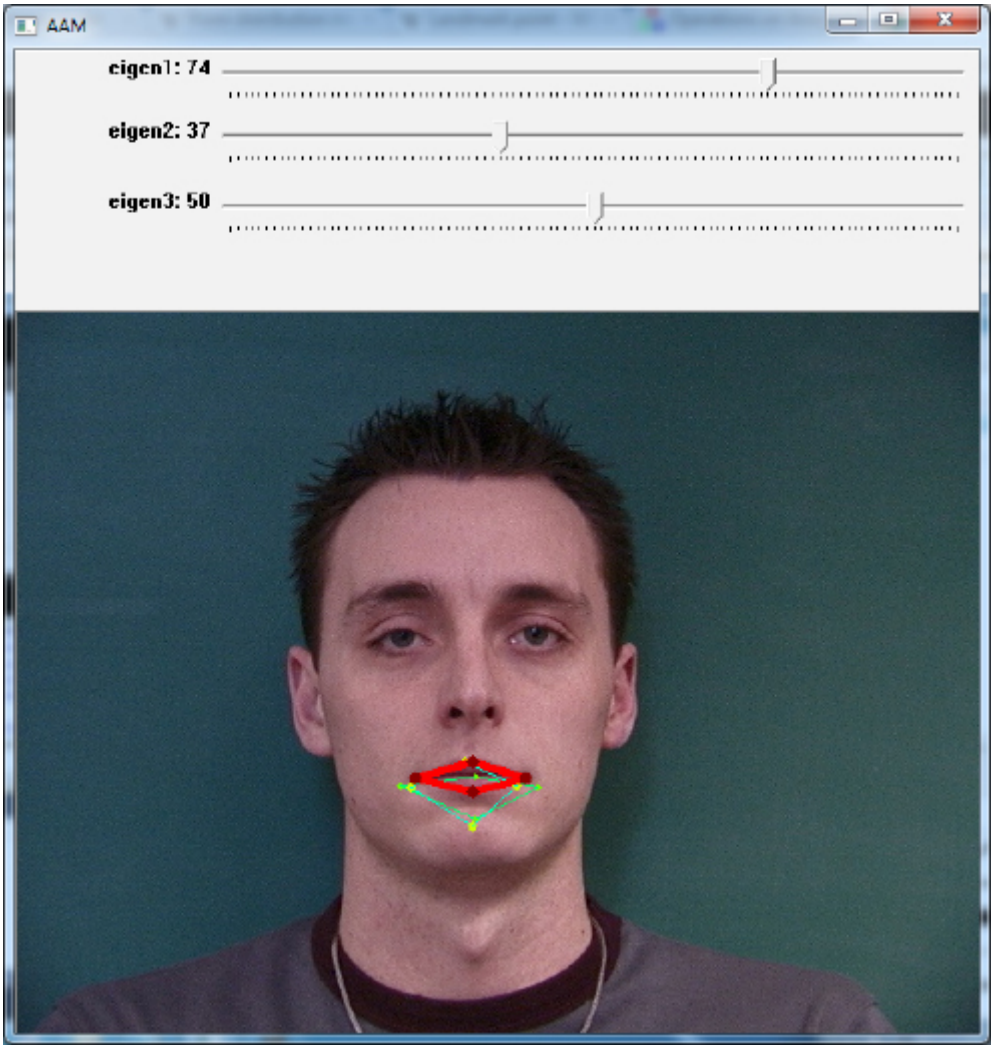

No Dependency
Shape Dependency
No Dependency
Shape Dependency
Specific
Generic

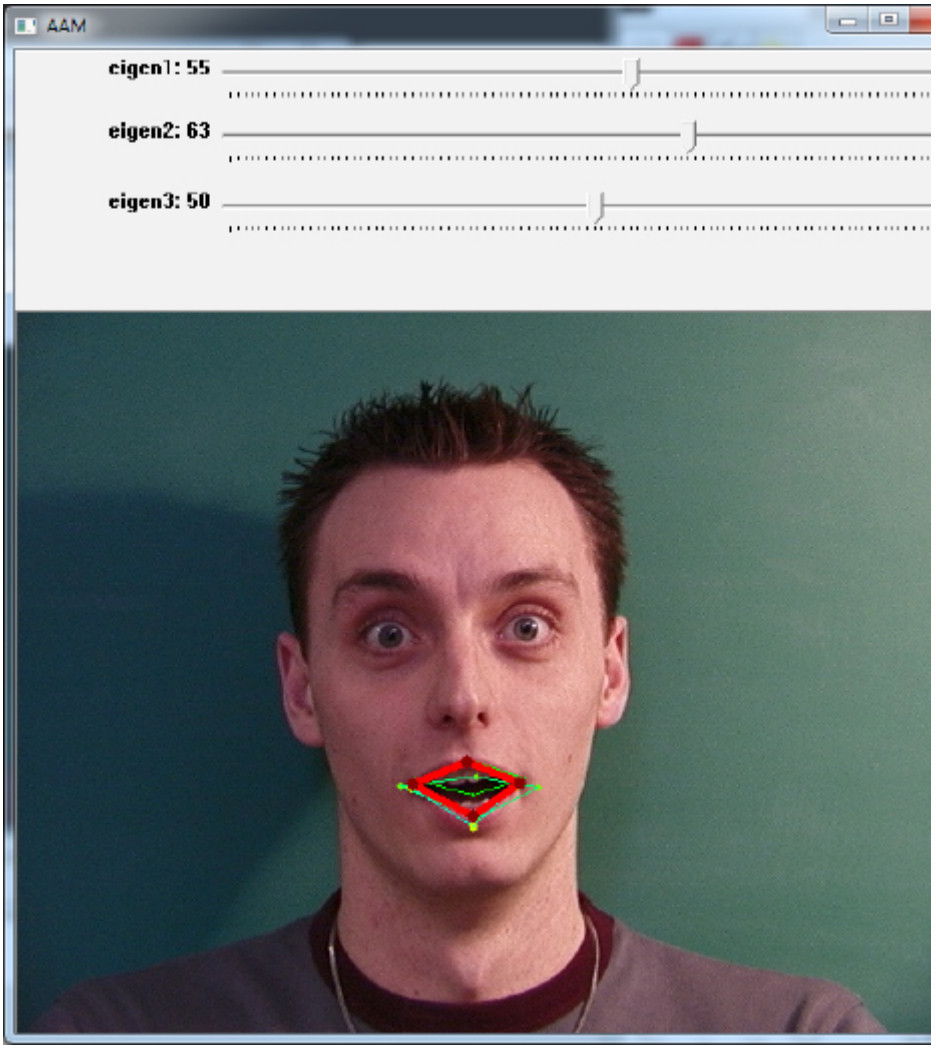
Chapter 5: 3D Head Pose Estimation Using AAM and POSIT

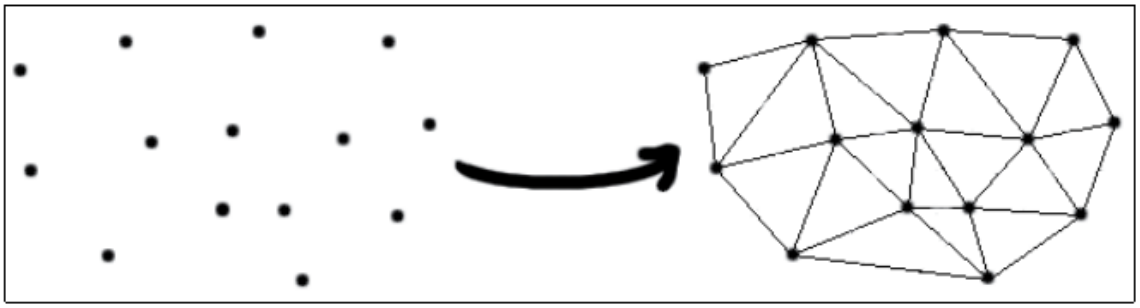


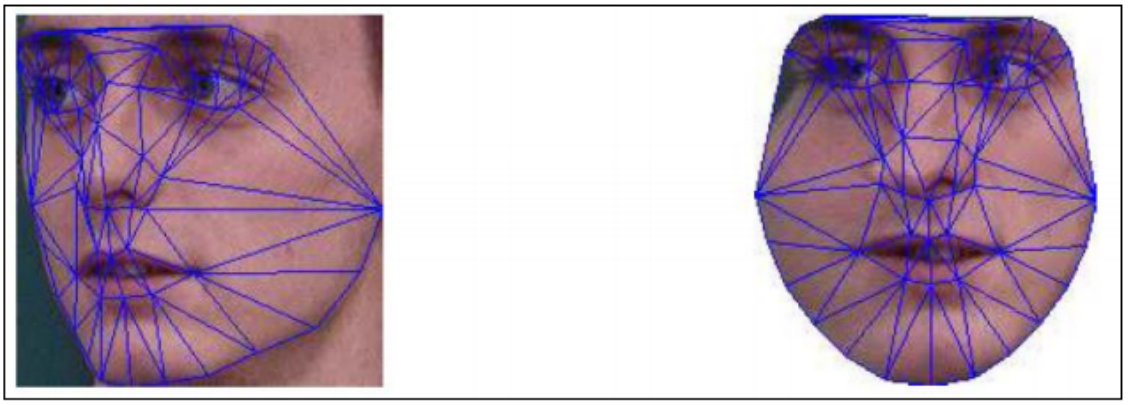



8
6
4
2
0
-2
Y
-4
-2
0
2
4
6
X


AAM
eigen1: 74
eigen2: 37
eigen3: 50


AAM
eigen1: 55
eigen2: 63
eigen3: 50





Pre-compute:

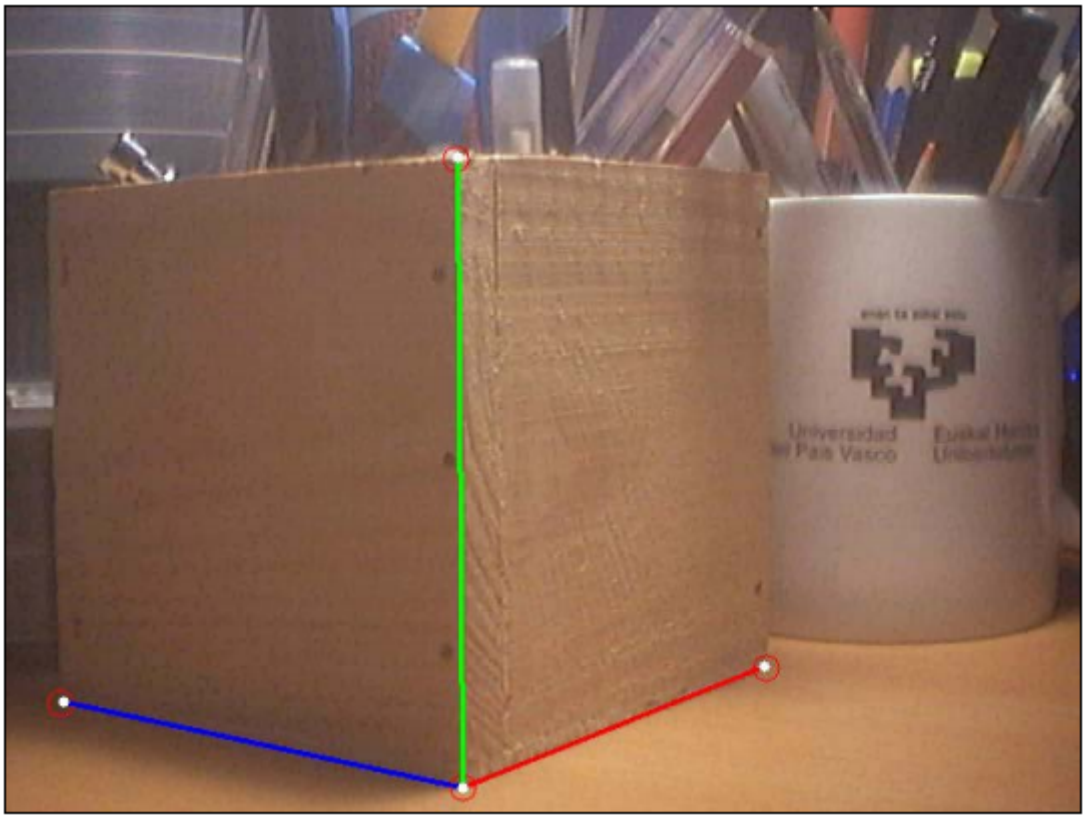
(3) Evaluate the gradient ∇A_0 of the template $A_0(\mathbf{x})$
(4) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$
(5) Compute the modified steepest descent images using Equation (41)
(6) Compute the Hessian matrix using modified steepest descent images

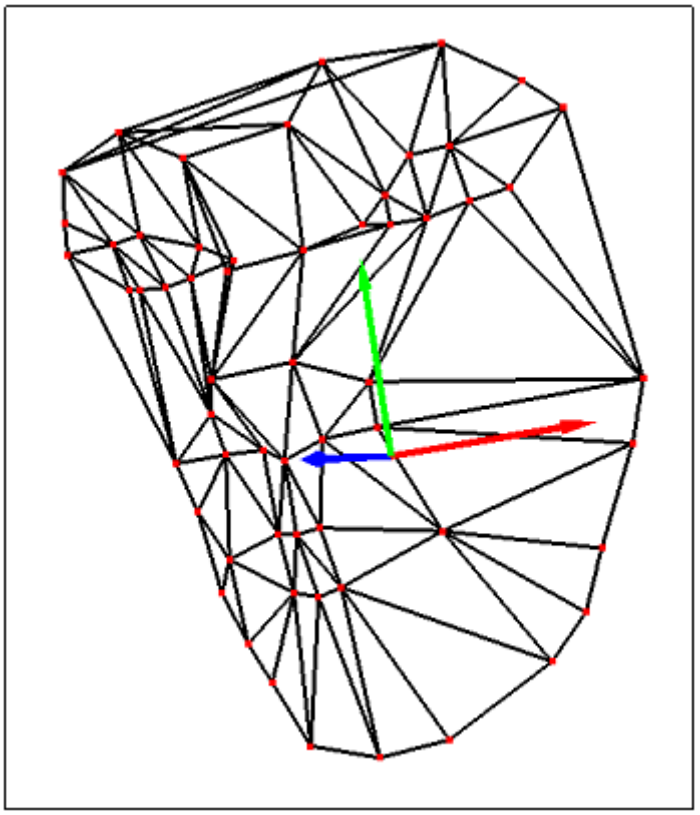
Iterate:

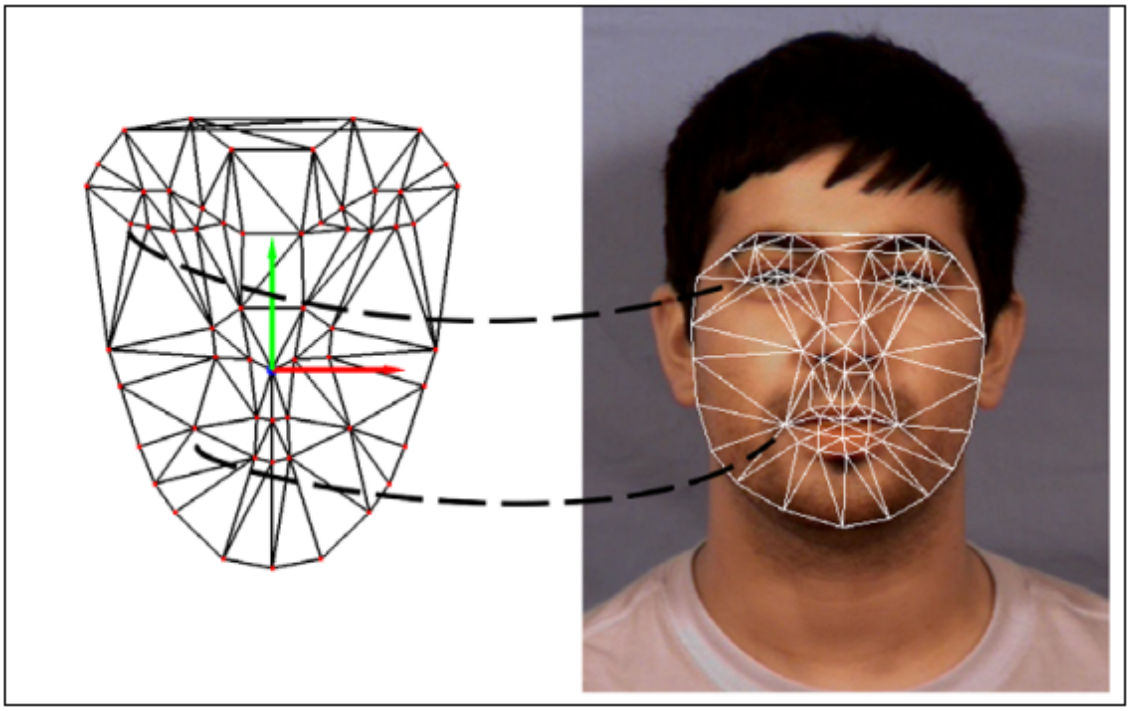
(1) Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
(2) Compute the error image $I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A_0(\mathbf{x})$
(7) Compute dot product of modified steepest descent images with error image
(8) Compute $\Delta \mathbf{p}$ by multiplying by inverse Hessian
(9) Update the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$

Post-computation:

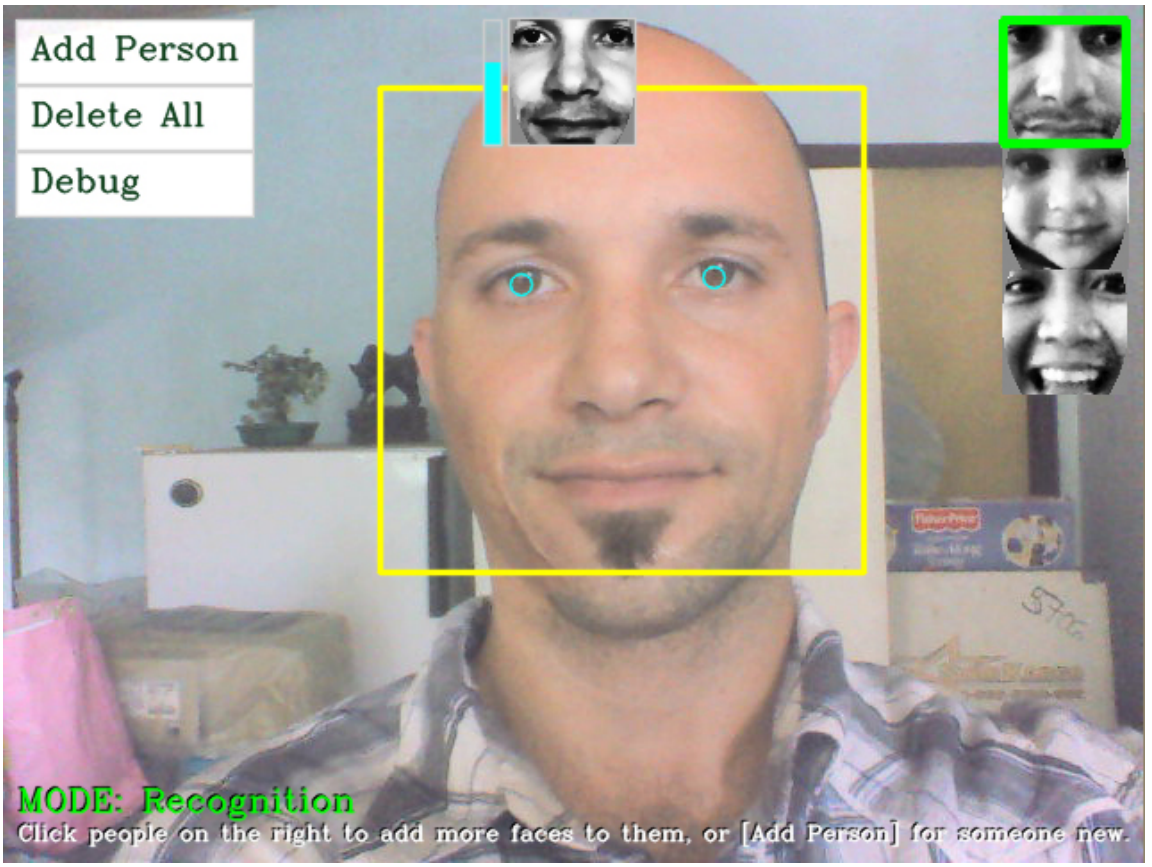
(10) Compute λ_i using Equation (40). [Optional step]

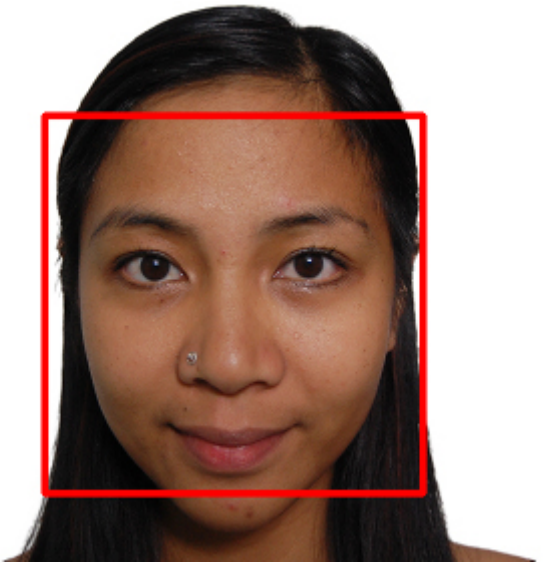

Universidad
del País Vasco

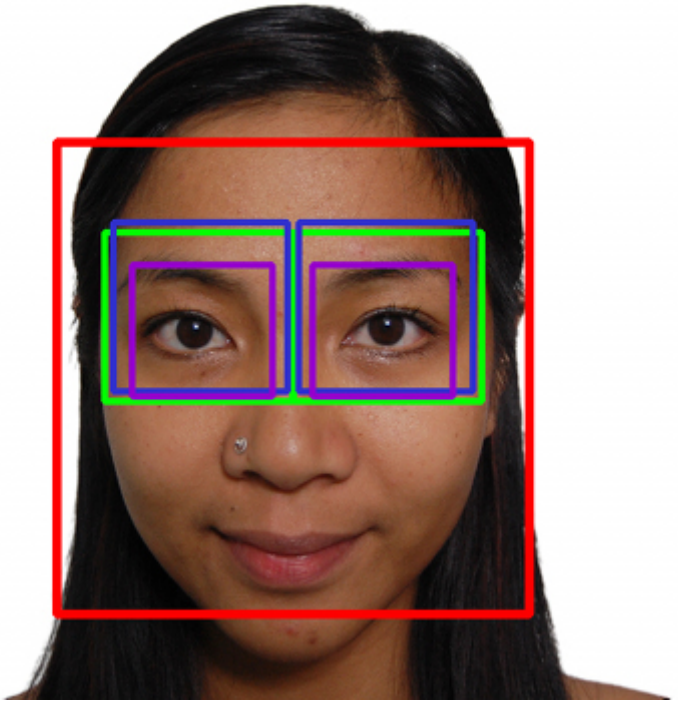


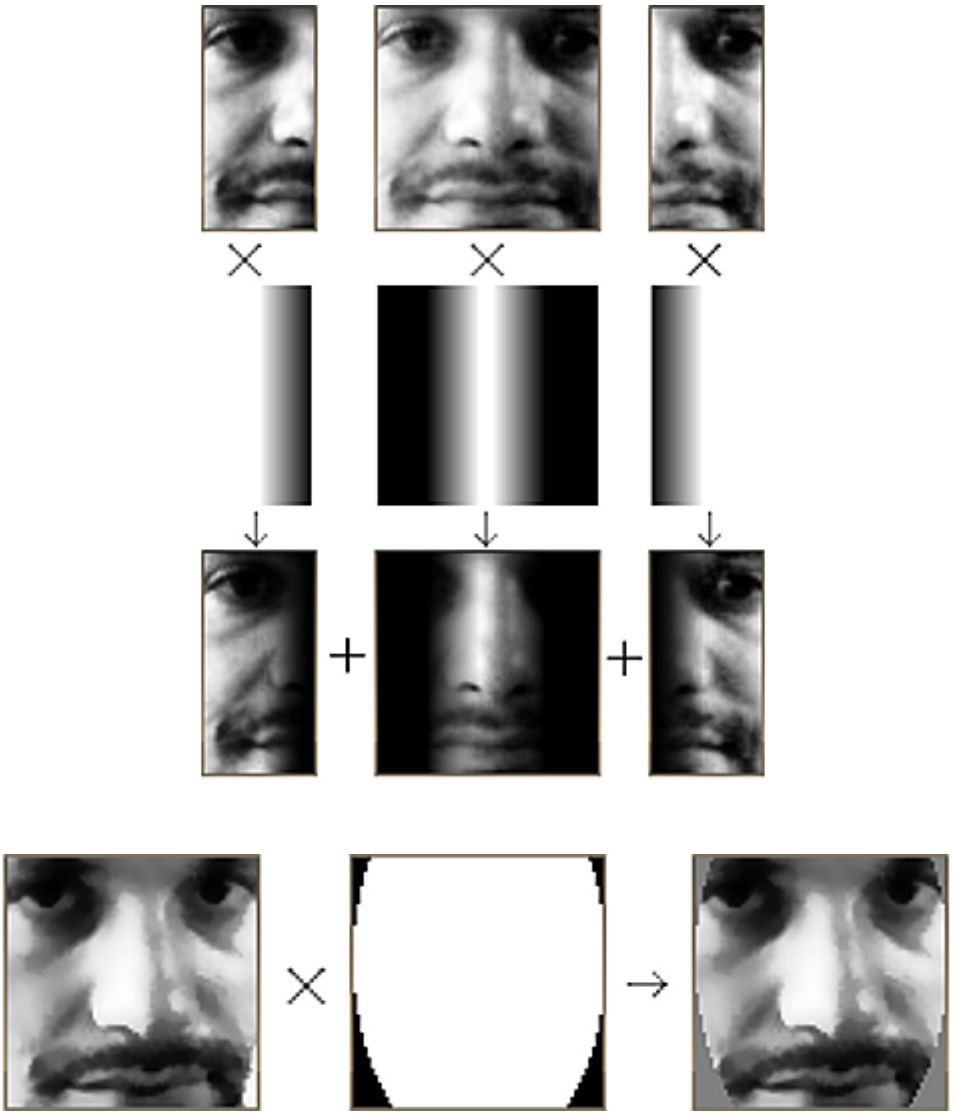


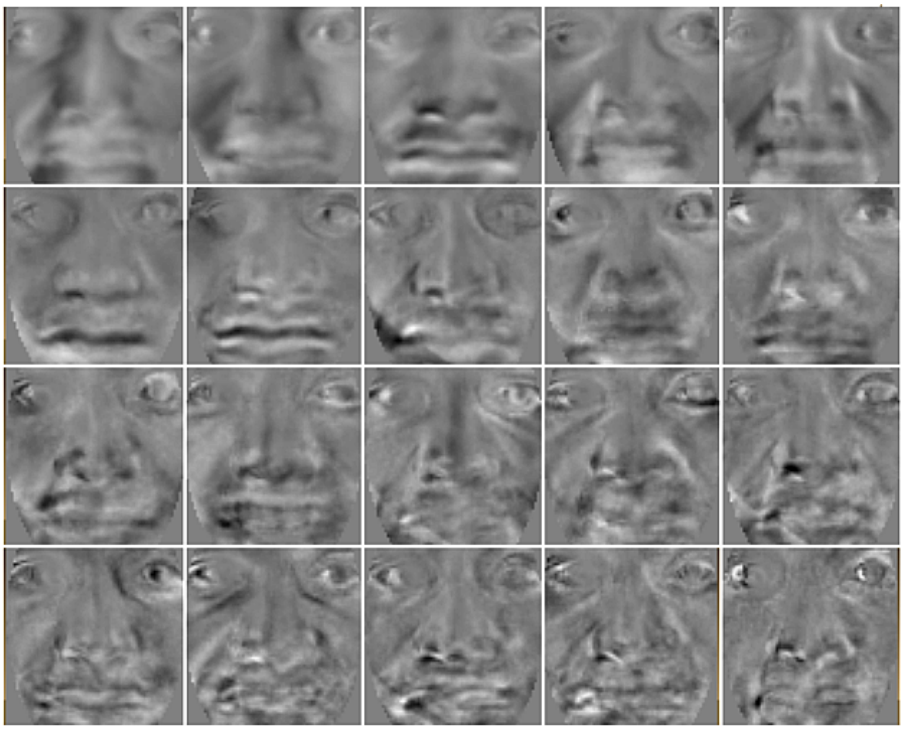
Chapter 6: Face Recognition Using Eigenfaces or Fisherfaces

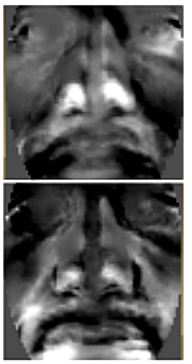


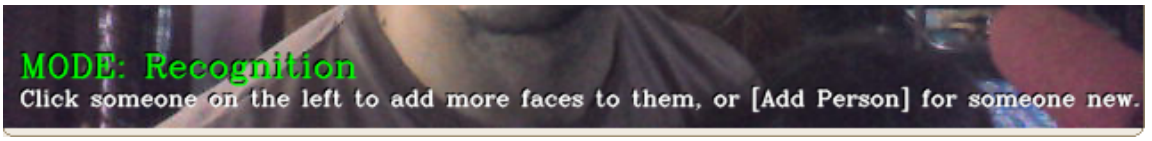




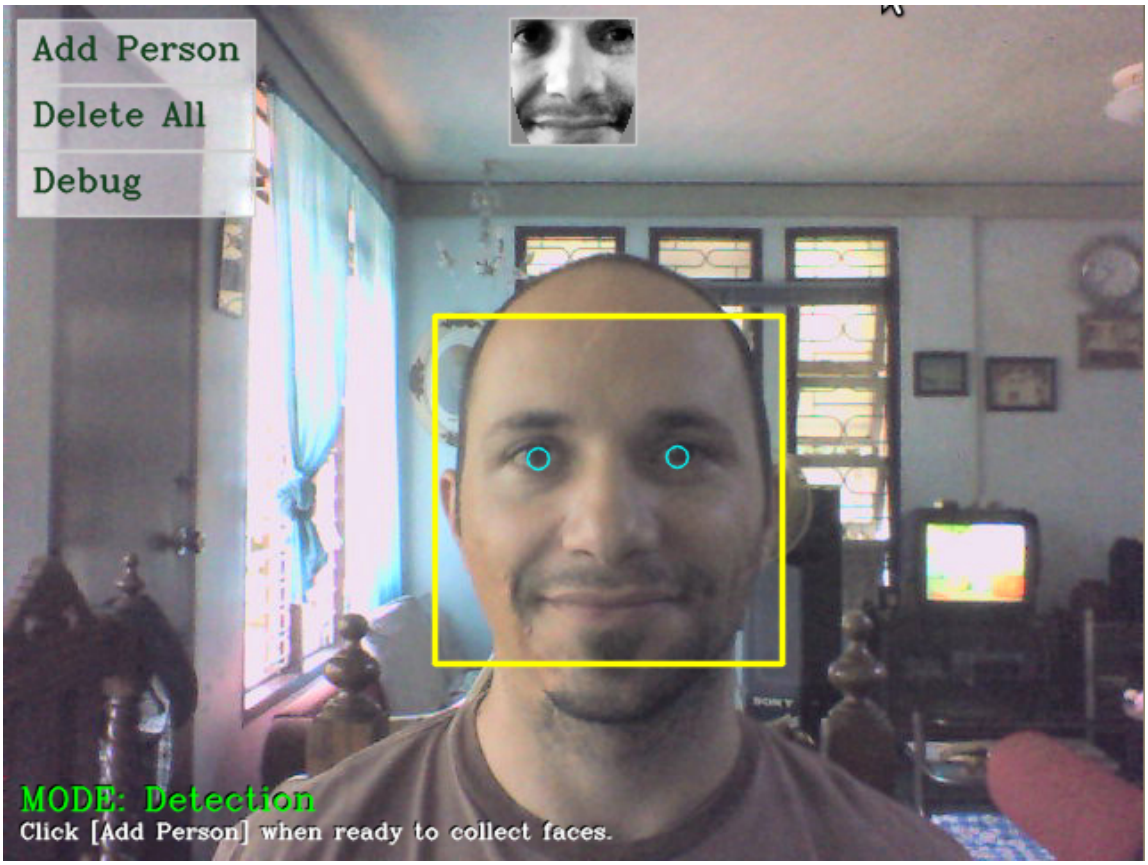


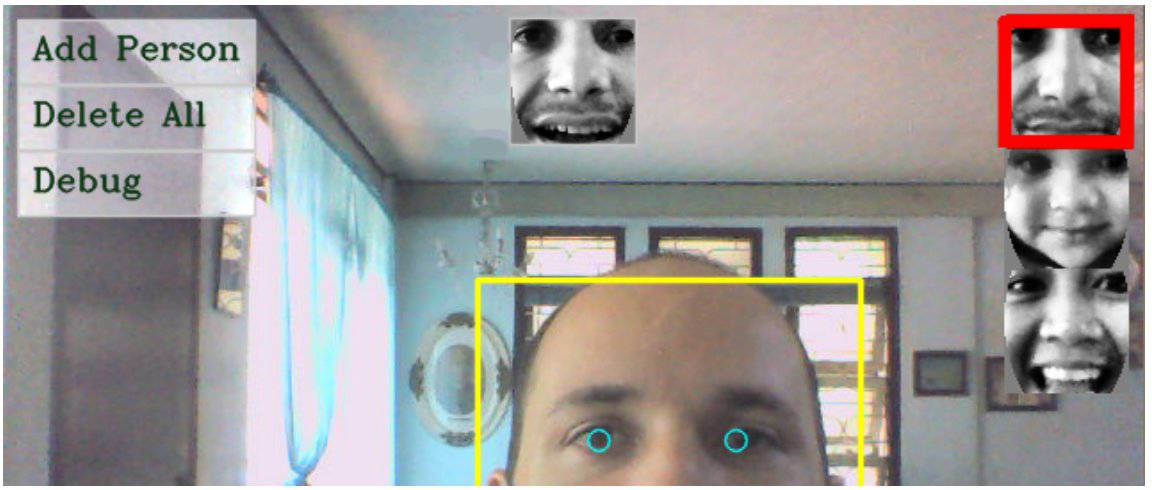





MODE: Recognition
Click someone on the left to add more faces to them, or [Add Person] for someone new.


Add Person
Delete All
Debug


Add Person
Delete All
Debug
MODE: Detection
Click [Add Person] when ready to collect faces.


Add Person
Delete All
Debug