A Cheap Portable Eye-Tracker Solution for Common Setups

Onur Ferhat, Fernando Vilariño

Computer Vision Center, Spain
Computer Science Dpt., Univ. Autònoma de Barcelona
oferhat@cvc.uab.es, fernando@cvc.uab.es

August 13, 2013

Outline

Problem Definition

- **Expensive** (€,€€€)
- **Expertise** required for scientists
- Not accessible to **regular computer users**
- **ADVANTAGE:** High accuracy (**0.1°-1°**) and high refresh rates (up to **600 Hz**)

Problem Definition

- **Expensive** (€,€€€)
- **Expertise** required for scientists
- Not accessible to **regular computer users**
- **ADVANTAGE:** High accuracy (**0.1°-1°**) and high refresh rates (up to **600 Hz**)

Problem Definition

- **Expensive** (€,€€€)
- **Expertise** required for scientists
- Not accessible to **regular computer users**
- **ADVANTAGE:** High accuracy (**0.1°-1°**) and high refresh rates (up to **600 Hz**)

Images ©Tobii Technology.

Problem Definition

- **Expensive** (€,€€€)
- **Expertise** required for scientists
- Not accessible to **regular computer users**
- **ADVANTAGE:** High accuracy (**0.1°-1°**) and high refresh rates (up to **600 Hz**)

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Our Aim

Building an eye-tracker that is:

- Cheap
- Easy to use
- Scalable
- Easy to build with custom components
- Compatible with a variety of operating systems
- Accurate and robust

Use Cases

Such a system can be used in:

- **Remote** eye-tracking studies [1]
- **Parallel experiments** in a laboratory
- Applications in **mobile** devices
- **Non-critical applications** such as games

[1] A. Bojko, "The Truth About Webcam Eye Tracking"

Use Cases

Such a system can be used in:

- **Remote** eye-tracking studies [1]
- **Parallel experiments** in a laboratory
- Applications in **mobile** devices
- **Non-critical applications** such as games

[1] A. Bojko, "The Truth About Webcam Eye Tracking"

Use Cases

Such a system can be used in:

- **Remote** eye-tracking studies [1]
- **Parallel experiments** in a laboratory
- Applications in **mobile** devices
- **Non-critical applications** such as games

[1] A. Bojko, "The Truth About Webcam Eye Tracking"

Use Cases

Such a system can be used in:

- **Remote** eye-tracking studies [1]
- **Parallel experiments** in a laboratory
- Applications in **mobile** devices
- **Non-critical applications** such as games

[1] A. Bojko, "The Truth About Webcam Eye Tracking"

Use Cases

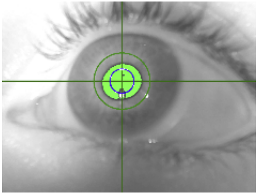
Such a system can be used in:

- **Remote** eye-tracking studies [1]
- **Parallel experiments** in a laboratory
- Applications in **mobile** devices
- **Non-critical applications** such as games

[1] A. Bojko, "The Truth About Webcam Eye Tracking"

Techniques (By Camera Type) [2]

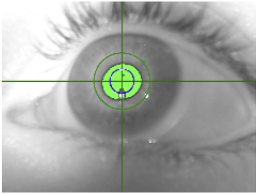
- **Infrared (IR):** IR methods use IR lights and cameras. Make use of the reflections of lights on the cornea.



[2] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze." IEEE TPAMI, vol. 32, no. 3, pp. 478–500, 2010.

Techniques (By Camera Type) [2]

- **Infrared (IR):** IR methods use IR lights and cameras. Make use of the reflections of lights on the cornea.



- **Visible light:** Use regular cameras and no additional light as part of the algorithm.

Image ©ITU Gaze Group.

[2] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze." IEEE TPAMI, vol. 32, no. 3, pp. 478–500, 2010.

Techniques (By Methodology)

- **Model based:** Fit a 2D/3D model to the eye/face. Then calculate the gaze point geometrically.

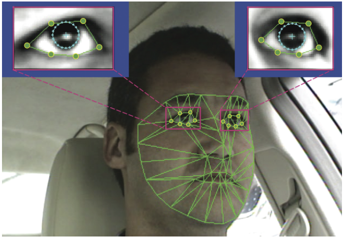
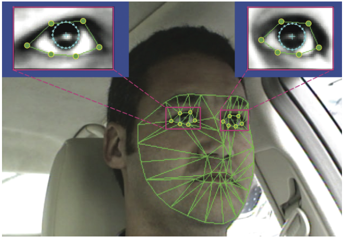


Image ©Takahiro Ishikawa.

Techniques (By Methodology)

- **Model based:** Fit a 2D/3D model to the eye/face. Then calculate the gaze point geometrically.



- **Appearance based:** Directly map the image of the eyes to the gaze point. Mostly neural networks (NN) are trained with image pixels as input and gaze point as output.

Open Source Eye-Tracker Software

There exist several open source options:

- **ITU Gaze Tracker** from IT Univ. of Copenhagen
  Uses IR cameras and IR light sources, camera image contains only an eye
- **Opengazer** from Univ. of Cambridge
  Uses regular webcams, no special lighting, camera image contains the entire face
- **Others:** EyeWriter, openEyes, Neural Network Eye Tracker (NNET), etc. [3]

[3] COGAIN, http://wiki.cogain.info/index.php/Links

Open Source Eye-Tracker Software

There exist several open source options:

- **ITU Gaze Tracker** from IT Univ. of Copenhagen
  Uses IR cameras and IR light sources, camera image contains only an eye
- **Opengazer** from Univ. of Cambridge
  Uses regular webcams, no special lighting, camera image contains the entire face
- **Others:** EyeWriter, openEyes, Neural Network Eye Tracker (NNET), etc. [3]

[3] COGAIN, http://wiki.cogain.info/index.php/Links

Open Source Eye-Tracker Software

There exist several open source options:

- **ITU Gaze Tracker** from IT Univ. of Copenhagen
  Uses IR cameras and IR light sources, camera image contains only an eye
- **Opengazer** from Univ. of Cambridge
  Uses regular webcams, no special lighting, camera image contains the entire face
- **Others:** EyeWriter, openEyes, Neural Network Eye Tracker (NNET), etc. [3]

[3] COGAIN, http://wiki.cogain.info/index.php/Links

Open Source Eye-Tracker Software

There exist several open source options:

- **ITU Gaze Tracker** from IT Univ. of Copenhagen
  Uses IR cameras and IR light sources, camera image contains only an eye
- **Opengazer** from Univ. of Cambridge
  Uses regular webcams, no special lighting, camera image contains the entire face
- **Others:** EyeWriter, openEyes, Neural Network Eye Tracker (NNET), etc. [3]

[3] COGAIN, http://wiki.cogain.info/index.php/Links

An Eye-Tracking System for a Single Webcam Setup

Hardware baseline:

- A regular webcam and a display

An Eye-Tracking System for a Single Webcam Setup

Hardware baseline:

- A regular webcam and a display

Software structure:

- Based on Opengazer [4], works under natural light
- 5 components modified
  - Point selection
  - Point tracking
  - Blink detection
  - Calibration
  - Estimation

[4] P. Zieliński, http://www.inference.phy.cam.ac.uk/opengazer/

An Eye-Tracking System for a Single Webcam Setup

Hardware baseline:

- A regular webcam and a display

Software structure:

- Based on Opengazer [4], works under natural light
- 5 components modified
  - Point selection
  - Point tracking
  - Blink detection
  - Calibration
  - Estimation

Hardware for portable eye-tracker:

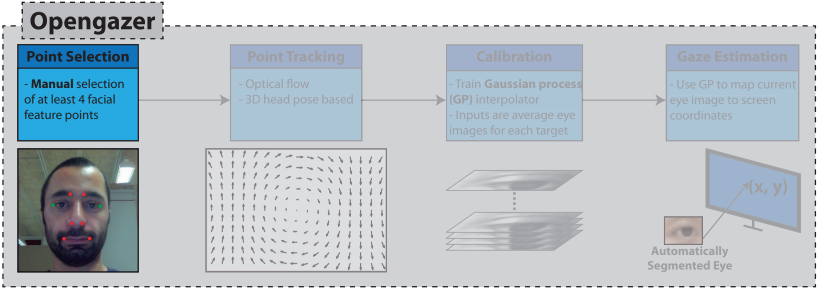
- Raspberry Pi, webcam and accessories (SD card, cables)

[4] P. Zieliński, http://www.inference.phy.cam.ac.uk/opengazer/

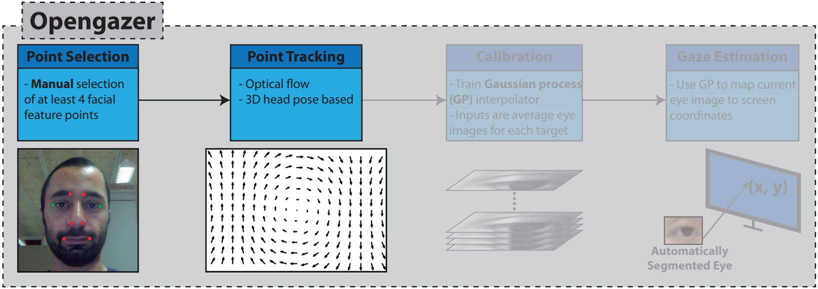
Video

[EYE-TRACKER VIDEO]

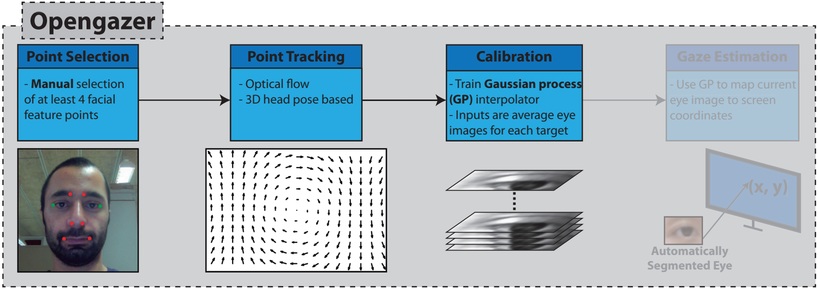
An Eye-Tracking System for a Single Webcam Setup



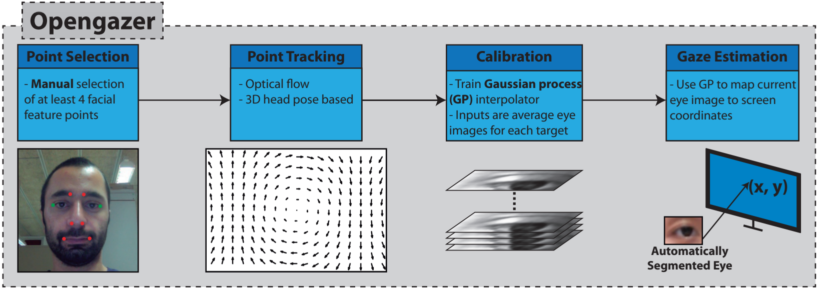
An Eye-Tracking System for a Single Webcam Setup



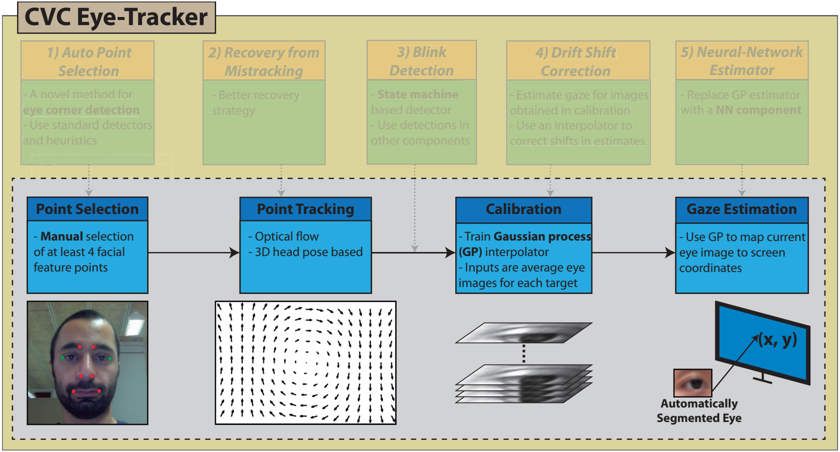
An Eye-Tracking System for a Single Webcam Setup



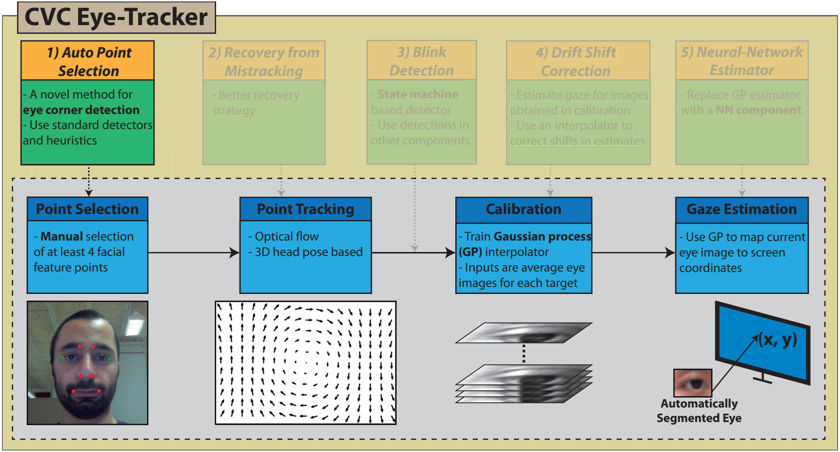
An Eye-Tracking System for a Single Webcam Setup



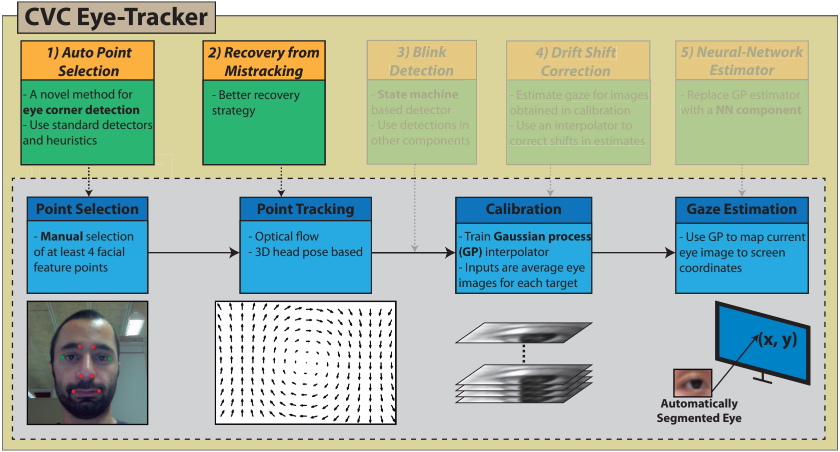
An Eye-Tracking System for a Single Webcam Setup



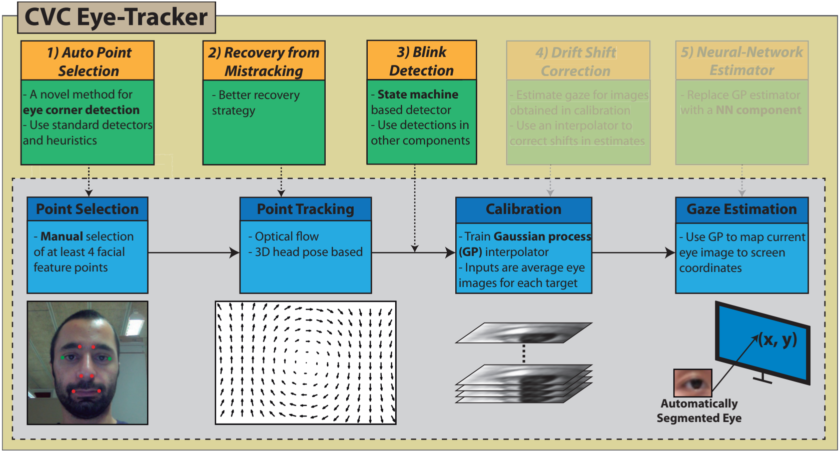
An Eye-Tracking System for a Single Webcam Setup



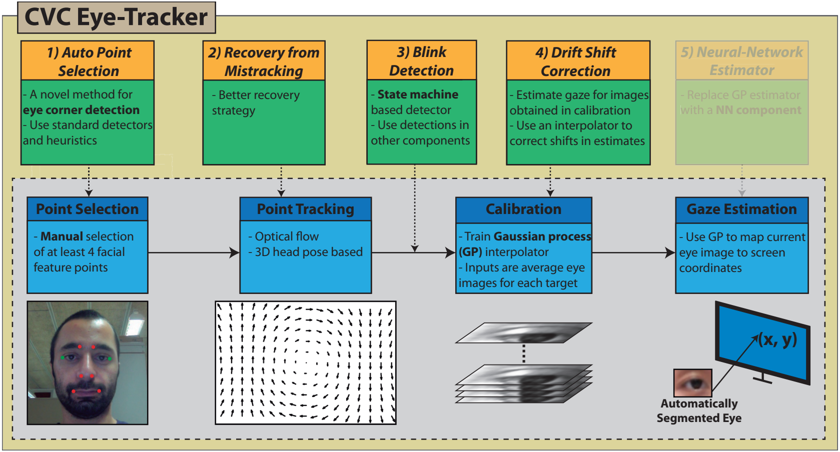
An Eye-Tracking System for a Single Webcam Setup



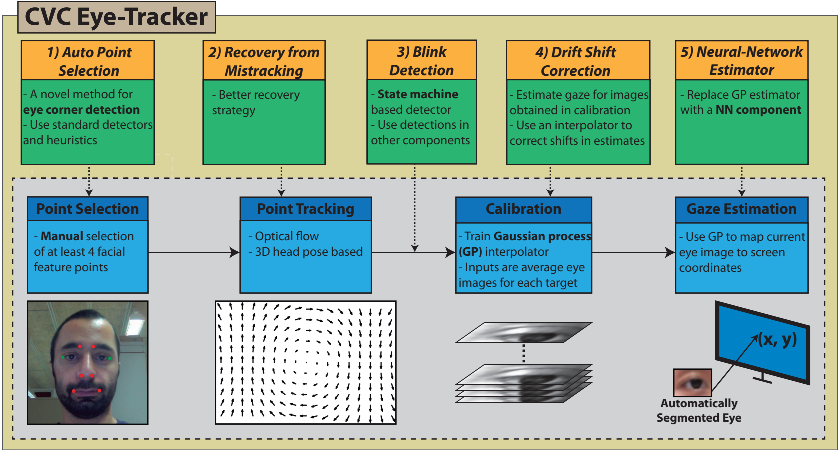
An Eye-Tracking System for a Single Webcam Setup



An Eye-Tracking System for a Single Webcam Setup

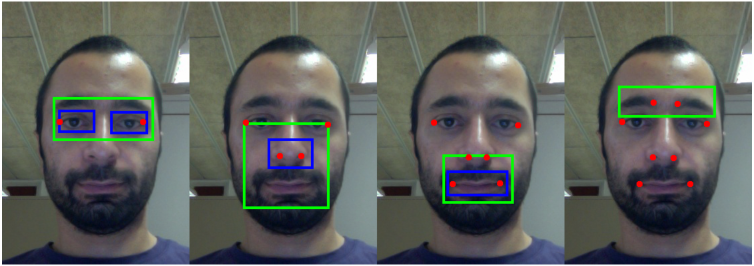


An Eye-Tracking System for a Single Webcam Setup



1) Automatic Point Selection

- Based on Haar cascades [5] (detectors for eye, nose, mouth)
- Use of geometrical heuristics (e.g. location of nosetips inside the detected nose rectangle)
- Novel method for fast eye-corner detection



[5] M. Castrillon-Santana, http://mozart.dis.ulpgc.es/Gias/modesto.html?lang=0

2) Tracking and 3) Blink Detection

- Improved strategies for point tracking and recovery from mistracking
  - In case of mistracking, use the 3D head pose based estimation
- A state-machine based blink detector
  - No estimation during blinks
  - Skip blinking frames during calibration

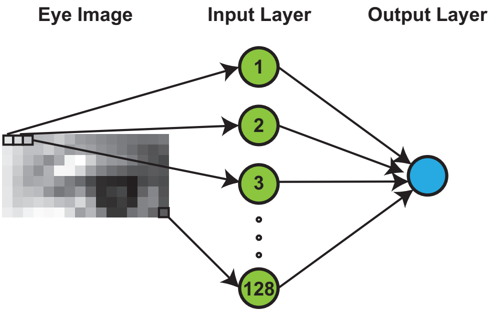
4) Calibration Error Correction

A drift shift correction method for improving training:

1. After calibration, calculate the gaze estimations for images acquired during calibration
2. Measure the errors between the average estimation for a target and actual target coordinates
3. Train a multivariate interpolator to correct these shifts
4. During testing, correct the estimations using the interpolator

5) Neural Network Estimator

Implement a neural network based estimator [6]:

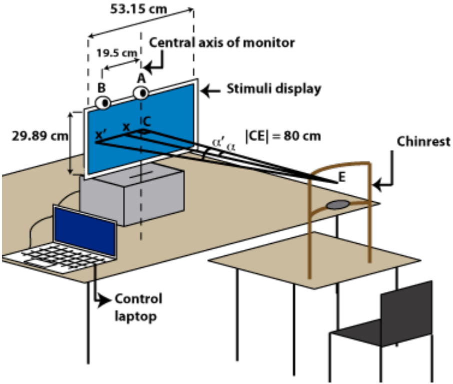


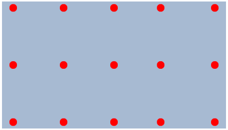
- Eye image resized to 16x8 pixels
- 4 networks for 2 eyes and 2 different outputs (X and Y coordinates)

[6] O. V. Komogortsev, http://cs.txstate.edu/ok11/nnet.html

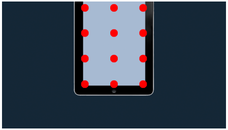
Experimental Setup

Components of the experimental setup and target point locations:





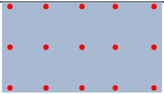
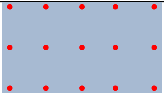
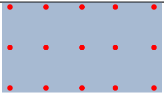
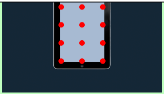
Entire screen (15 points)



iPad screen (12 points)

Setup Variations

Test for 4 different cases:

Setup	Distance	Camera	Chinrest	Target
STANDARD	$80cm$	Center		
EXTREME	$80cm$	Left		
CHINREST	$80cm$	Center	Yes	
iPAD	$40cm$	Center		

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

- **ORIG:** Original Opengazer + auto point selection

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

- **ORIG:** Original Opengazer + auto point selection
- **2-EYE:** Previous + estimation using 2 eyes

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

- **ORIG:** Original Opengazer + auto point selection
- **2-EYE:** Previous + estimation using 2 eyes
- **TRACK:** Previous + tracking improvements

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

- **ORIG:** Original Opengazer + auto point selection
- **2-EYE:** Previous + estimation using 2 eyes
- **TRACK:** Previous + tracking improvements
- **BLINK:** Previous + excluding blinks during calibration

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

- **ORIG:** Original Opengazer + auto point selection
- **2-EYE:** Previous + estimation using 2 eyes
- **TRACK:** Previous + tracking improvements
- **BLINK:** Previous + excluding blinks during calibration
- **CORR:** Previous + calibration error correction

STANDARD setup

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**1.82** (1.40)	95	**1.56** (0.81)	80
2-EYE	1.56 (1.28)	82	1.48 (0.86)	75
TRACK	1.73 (1.37)	91	1.58 (0.78)	81
BLINK	1.73 (1.37)	91	1.58 (0.76)	80
CORR	**1.62** (1.40)	85	**1.50** (0.77)	76
NN	5.12 (2.26)	266	2.26 (0.92)	115

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
ORIG	**2.07** (1.69)	109	**1.73** (1.04)	88
2-EYE	1.94 (2.14)	101	1.67 (1.07)	85
TRACK	1.77 (1.52)	93	1.50 (0.80)	76
BLINK	1.78 (1.53)	93	1.49 (0.80)	76
CORR	**1.68** (1.56)	88	**1.43** (0.80)	73
NN	5.33 (2.83)	277	2.09 (0.89)	107

1280×720 camera

- **ORIG:** Original Opengazer + auto point selection
- **2-EYE:** Previous + estimation using 2 eyes
- **TRACK:** Previous + tracking improvements
- **BLINK:** Previous + excluding blinks during calibration
- **CORR:** Previous + calibration error correction
- **NN:** Previous + neural network estimator

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

1280×720 camera

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

1280×720 camera

Conclusions:

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

1280×720 camera

Conclusions:

- Decrease in errors by **18%** hor. and **17%** ver. (std. setup)

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	°(σ)	Px	°(σ)	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

1280×720 camera

Conclusions:

- Decrease in errors by **18%** hor. and **17%** ver. (std. setup)
- Higher **resolution** camera almost always improves results

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

1280×720 camera

Conclusions:

- Decrease in errors by **18%** hor. and **17%** ver. (std. setup)
- Higher **resolution** camera almost always improves results
- Camera **placement** has an effect on the performance

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

1280×720 camera

Conclusions:

- Decrease in errors by **18%** hor. and **17%** ver. (std. setup)
- Higher **resolution** camera almost always improves results
- Camera **placement** has an effect on the performance
- **Chinrest** decreases the errors by **32%** horizontally

Comparison of Setups (CORR)

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.62 (1.40)	85	1.50 (0.77)	76
Extreme	1.88 (1.37)	98	1.78 (0.58)	91
Chinrest	1.25 (0.78)	66	1.53 (0.44)	78
iPad	2.49 (1.64)	64	2.08 (0.92)	54

640×480 camera

	Horizontal err.		Vertical err.	
	$^\circ(\sigma)$	Px	$^\circ(\sigma)$	Px
Standard	1.68 (1.56)	88	1.43 (0.80)	74
Extreme	1.76 (1.15)	92	1.78 (0.67)	91
Chinrest	1.14 (0.79)	60	1.36 (0.53)	69
iPad	2.15 (1.32)	55	2.02 (0.76)	52

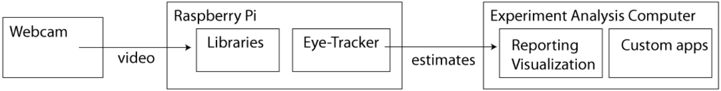
1280×720 camera

Conclusions:

- Decrease in errors by **18%** hor. and **17%** ver. (std. setup)
- Higher **resolution** camera almost always improves results
- Camera **placement** has an effect on the performance
- **Chinrest** decreases the errors by **32%** horizontally
- In **iPad** setup, pixel errors are smaller but angular errors appear larger

A Portable Eye-Tracker

Feasibility analysis of our system as a portable, stand-alone eye-tracker



- The video stream is processed by **Raspberry Pi**
- Estimates can be accessed in **real-time** (ethernet) or **offline** (text file)
- The **reporting** and **visualisation** tools run in the analysis computer
- The eye-tracker can estimate the gaze with ~**4 fps** in 640×480

Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)

Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)
- Built an eye-tracker with **1.62**$^\circ$ hor. **1.50**$^\circ$ ver. accuracy (for VGA quality)

Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)
- Built an eye-tracker with **1.62**° hor. **1.50**° ver. accuracy (for VGA quality)
  - Commercial models have **0.1°-1°** accuracy, however the price difference is huge (**thousands** of Euros vs. **70** Euros)

Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)
- Built an eye-tracker with **1.62**$^\circ$ hor. **1.50**$^\circ$ ver. accuracy (for VGA quality)
  - Commercial models have **0.1**$^\circ$**-1**$^\circ$ accuracy, however the price difference is huge (**thousands** of Euros vs. **70** Euros)
  - The achieved accuracy is useful in non-critical cases

Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)
- Built an eye-tracker with **1.62**° hor. **1.50**° ver. accuracy (for VGA quality)
  - Commercial models have **0.1**°**-1**° accuracy, however the price difference is huge (**thousands** of Euros vs. **70** Euros)
  - The achieved accuracy is useful in non-critical cases
- On a personal computer (Mac or Linux), the system runs with a frame rate of more than **15 fps**

Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)
- Built an eye-tracker with **1.62**$^\circ$ hor. **1.50**$^\circ$ ver. accuracy (for VGA quality)
  - Commercial models have **0.1**$^\circ$**-1**$^\circ$ accuracy, however the price difference is huge (**thousands** of Euros vs. **70** Euros)
  - The achieved accuracy is useful in non-critical cases
- On a personal computer (Mac or Linux), the system runs with a frame rate of more than **15 fps**
- The Raspberry Pi SD card image is ready for download (**http://bit.ly/cvc-raspberry**)

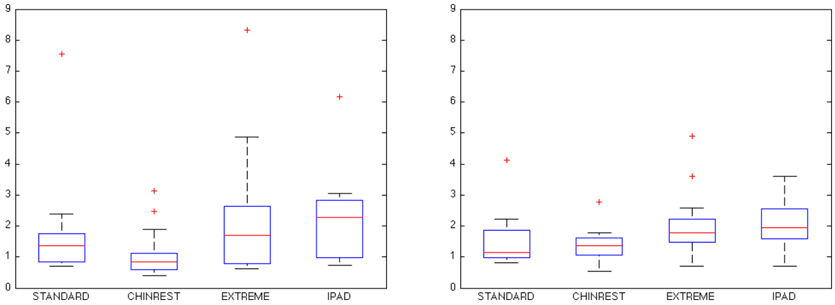
Conclusions

- Improved the state-of-the-art (**18%** hor. and **17%** ver.)
- Built an eye-tracker with **1.62**° hor. **1.50**° ver. accuracy (for VGA quality)
  - Commercial models have **0.1°-1°** accuracy, however the price difference is huge (**thousands** of Euros vs. **70** Euros)
  - The achieved accuracy is useful in non-critical cases
- On a personal computer (Mac or Linux), the system runs with a frame rate of more than **15 fps**
- The Raspberry Pi SD card image is ready for download (**http://bit.ly/cvc-raspberry**)
- The eye-tracker (Mac OS X and Linux) is ready for download (**http://bit.ly/cvc-eyetracker**)

The End

Thanks for your time!

Comparison of Subject Performances



Box plots for subject error rates horizontally (left) and vertically (right) under 1280×720 camera