

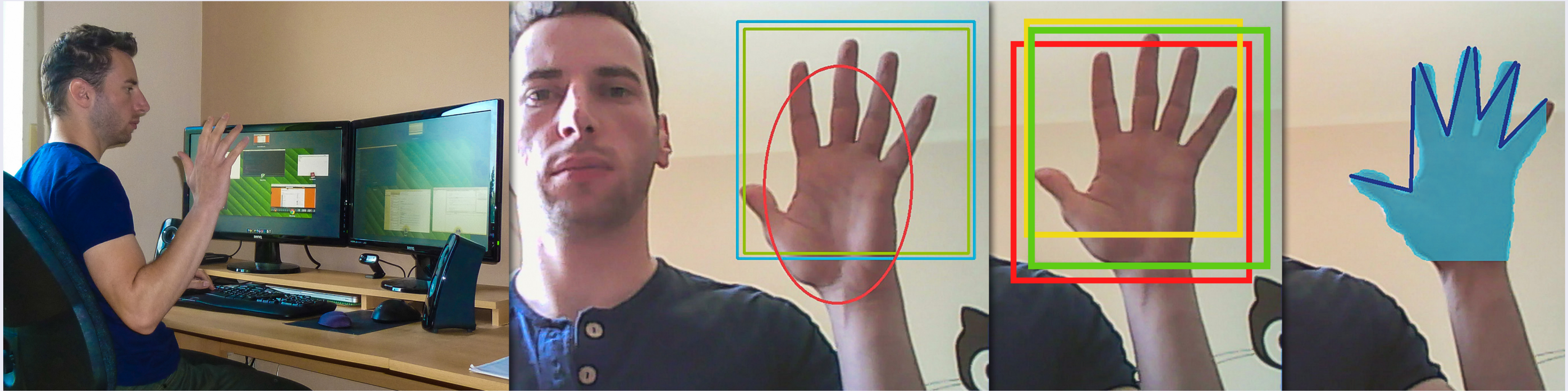
# Human-Computer Interaction Using Robust Gesture Recognition

Matthias Endler, Oleg Lobachev, and Michael Guthe

University Bayreuth

Jun. 4th, 2014

## Hand detection and tracking



From left to right: an overview of the setup; input frame from the webcam with Haar-detected bounding box, CAMSHIFT-fitted-ellipse, and bootstrapped-Shape-bounding box; Haar bootstrapping (happens only once, it is an initialization step); final result with hand tracked and fingers counted, used to control applications.

## Abstract

We present a detector cascade for robust real-time tracking of hand movements on consumer-level hardware. We adapt existing detectors to our setting: Haar, CAMSHIFT, shape detector, skin detector. We use *all* these detectors at once. A unified interface to all the detectors was introduced to facilitate our approach. Our main contributions are: first, utilization of *bootstrapping*: Haar bootstraps itself, then its results are used to bootstrap the other filters; second, the usage of temporal filtering for more robust detection and to remove outliers; third, we adapted the detectors for more robust hand detection. The input is an unaugmented live 2D stream from a webcam. The resulting system produces very robust results in real time. We evaluate both the robustness and the real-time capability.

## Gestures used as an input in games



## Benchmarking

Filter	Frames per second			
	mean	median	variance	min.
Haar	107.74	109.98	331.46	60.51
CAMSHIFT	394.4	403.9	1402.99	161.5
Shape	125.62	130.11	198.81	36.98
Skin	774.4	732.5	48210.8	157.7

## Combined execution time

We combined the *worst* possible frame rates for all methods – it could be only better in practice – and we observe that our implementation is definitely real-time capable: the median is over **45 fps** on a MacBook Pro with 2.4 GHz Intel Core 2 Duo / 4 GB / integrated camera. The software was written with Python 2.7 and OpenCV 2.4.6.

## Robustness

#	Background	Lighting conditions	Gesture	Speed	Result
1	Simple	Normal	Exposé	Slow	+
2	Simple	Normal	Exposé	Fast	+
3	Simple	Overexp.	Move	Slow	+
4	Simple	Overexp.	Exposé	Slow	+
5	Simple	Underexp.	Exposé	Slow	+
6	Skin-colored	Underexp. (Noise)	Exposé	Fast	+
7	Moving	Changing	Exposé	Slow	+
8	Reflections	Underexp.	Exposé	Fast	+
9	Reflections	Underexp.	Move	Slow	-
10	Reflections	Normal	Move	Slow	+

Reference videos available online.

## The *boxplot* for the combined execution time

