Real-Time Face Tracking with a PTZ Camera

Project for CS4243 – Computer Vision

Group 3:
Florian Huber
Tobias Pfaffelmoser
Sebastian Sichelschmidt

NUS
National University
of Singapore

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

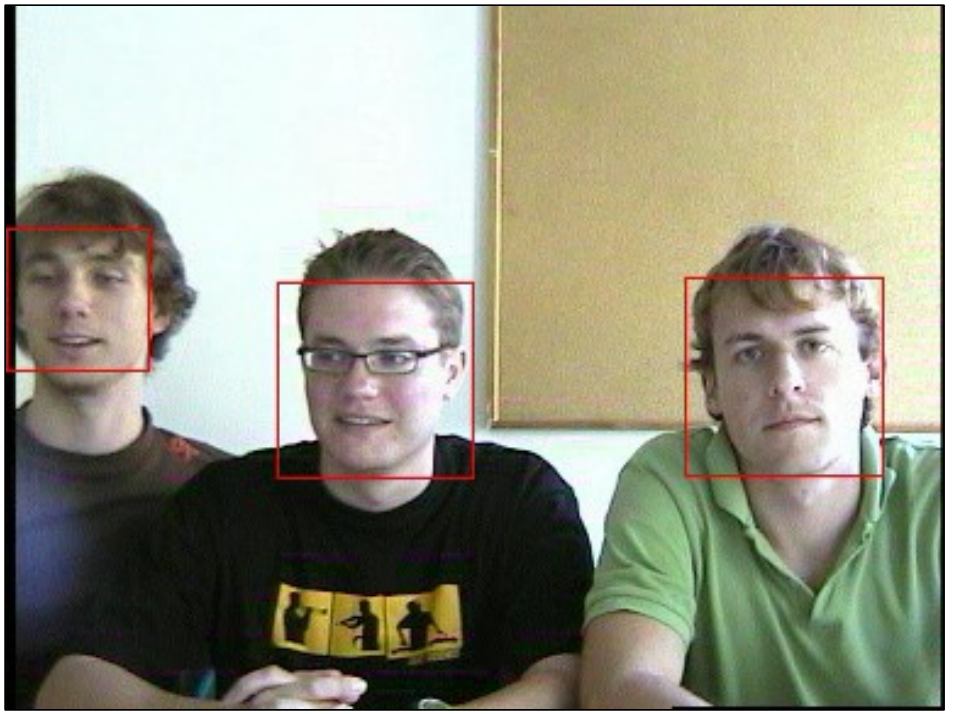
BERGISCHE
UNIVERSITÄT
WUPPERTAL

Outline

- Face Detection
- Histogram Generation
- Face Tracking
- PTZ Camera Control

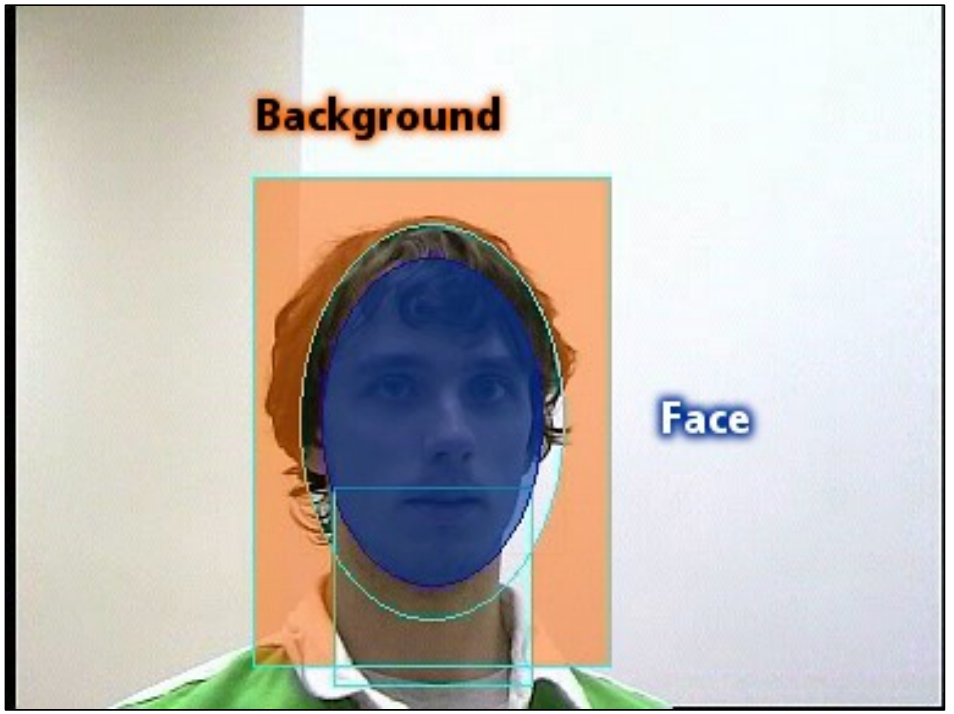
Face Detection

- Find all faces by OpenCV "*cvHaarDetectObjects*"
- Training data: cascade "haarcascade_frontalface_alt2.xml"
- Selection of face via mouse click
- Rectangular area as input for histogram generation



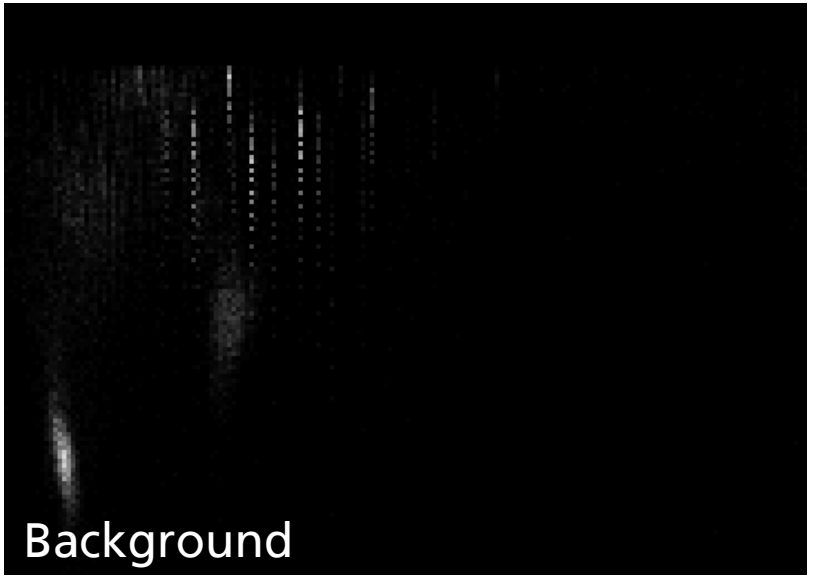
Histogram Generation

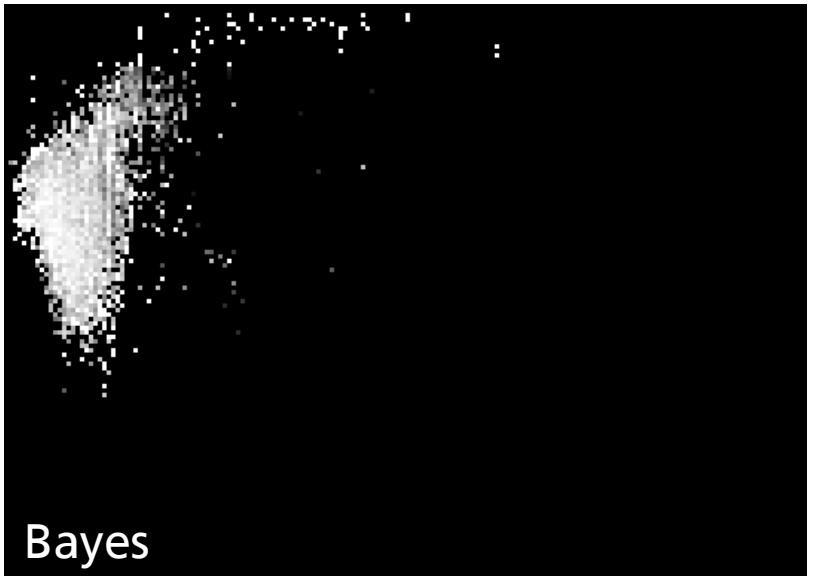
- HSV color space: lighting invariant tracking
- 2D histograms (hue/saturation) for face and background classification
- Histogram with face probability using Bayes Rule
- Dynamic update of histograms during tracking mode



$$P(\text{face} \mid c_{HS}) = \frac{P(c_{HS} \mid \text{face})P(\text{face})}{P(c_{HS})}$$

Histogram Generation



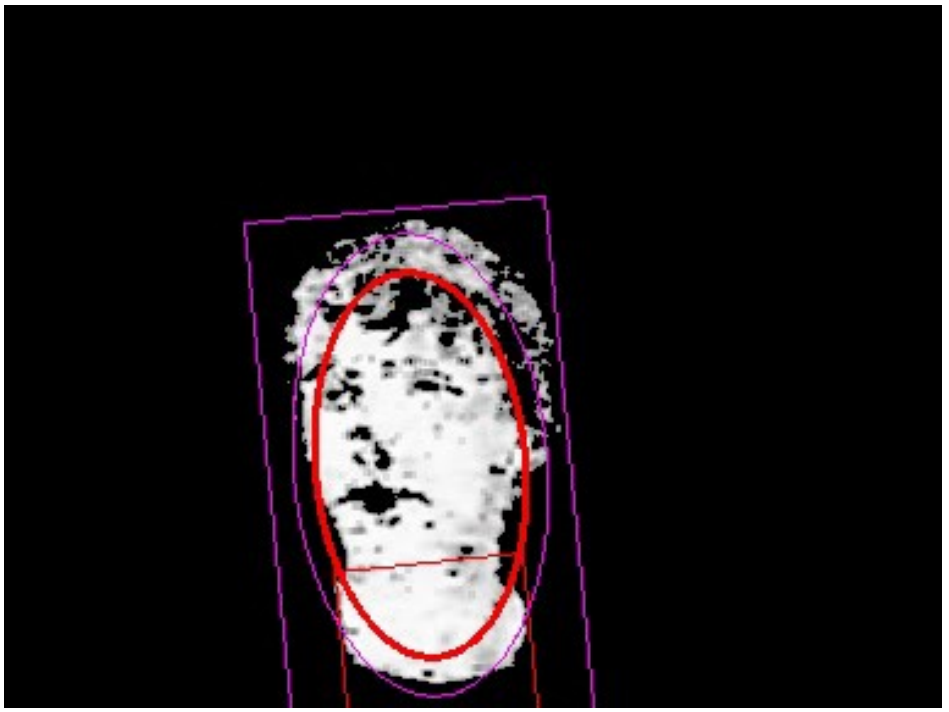


Saturation

Hue

Face Tracking

- *CAMShift*: Continuously Adaptive Mean Shift
- Lookup of face probabilities in *Bayes* Histogram
- Using mean for localization
- Computing face position, width and angle



PTZ Camera Control

- Accessing PTZ via asynchronous network sockets (HTTP)
- Estimating state (x, y, width, height, velocity) of tracked face with Kalman Filter
- Centering of face in rectangular area using pan and tilt
- Keep face height within certain constraints using zoom
- Lost-Face-Case: performing continuous pan with latest measured speed and direction

Project Milestones

- Environment setup (MS Platform SDK, MS DirectX SDK, AXIS SDK, OpenCV)
- Linking DirectShow, OpenCV and AXIS Cam (Iot's of trouble!)
- Face Detection experiments with Haar Detector (OpenCV vs. Torch3Vision)
- Reverse Engineering of camera and motion parameters
- Face Tracking experiments: CAMShift, BayesShift, Condensation
- Real-Time optimization
- Robustness improvements