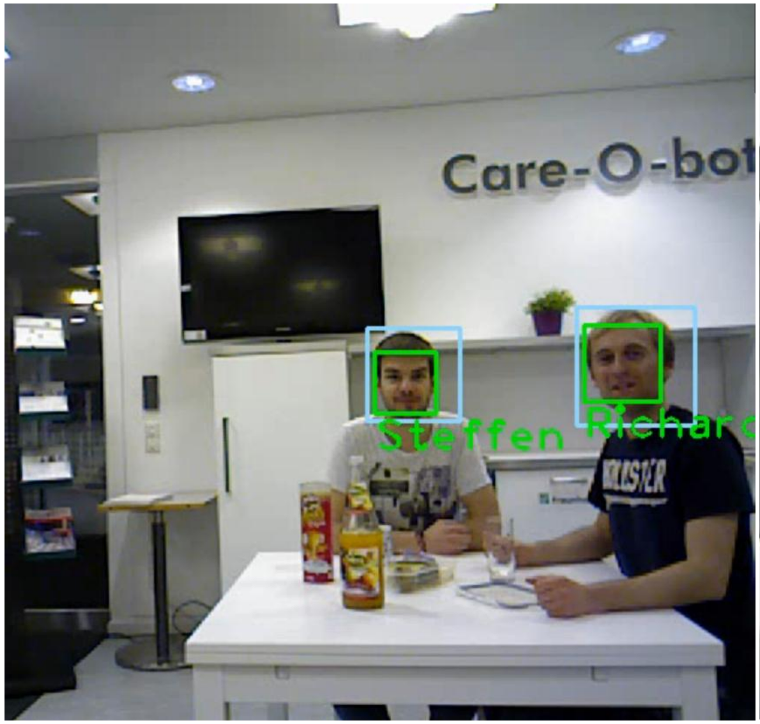
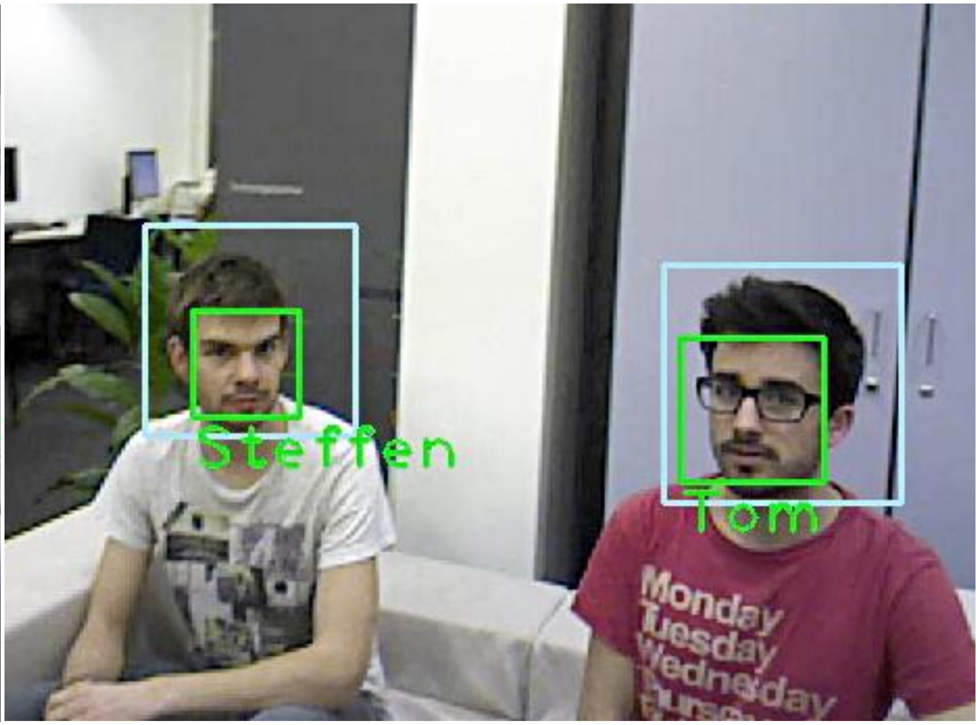
HI RICHARD - PERSONALIZE YOUR ROBOT WITH THE COB_PEOPLE_PERCEPTION STACK

Richard Bormann, Thomas Zwölfer, Jan Fischer

Fraunhofer-Institute for Manufacturing Engineering and Automation IPA





Fraunhofer IPA

Introduction

Fraunhofer
IPA

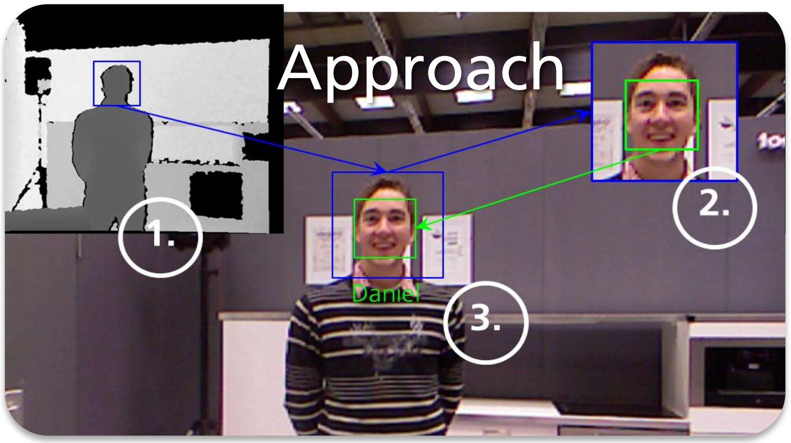
Introduction

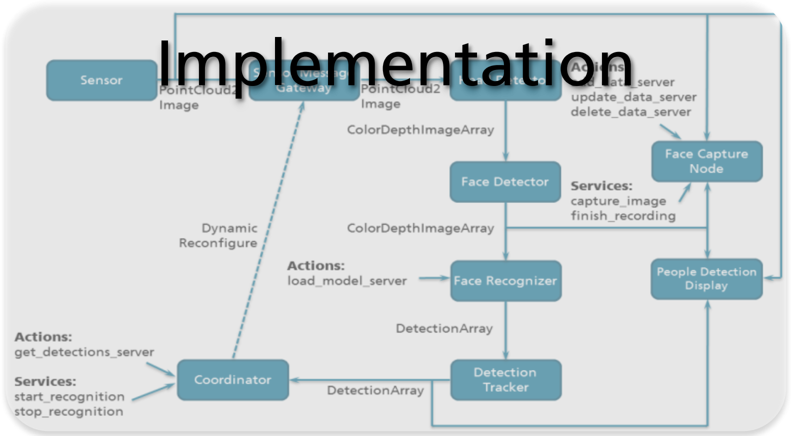
- personalized greeting
- appropriate behavior according to the user's preferences
- user centered services, reminders, etc.

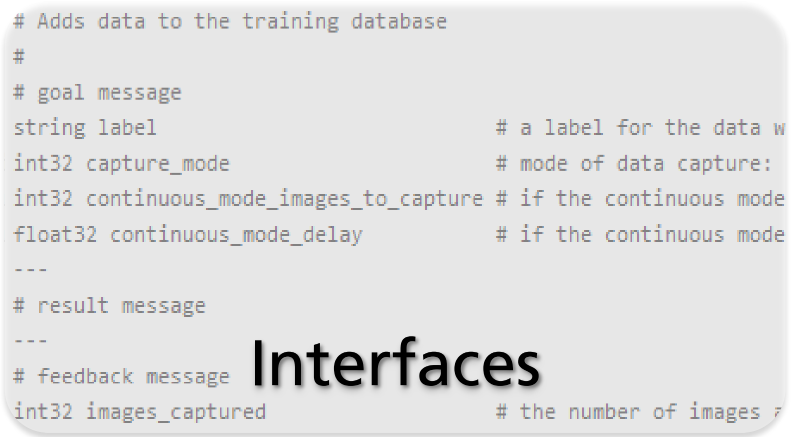
Fraunhofer IPA

Outline

This talk explains the cob_people_perception stack to detect and identify people. Details on implementation, interfaces and usage will be provided.





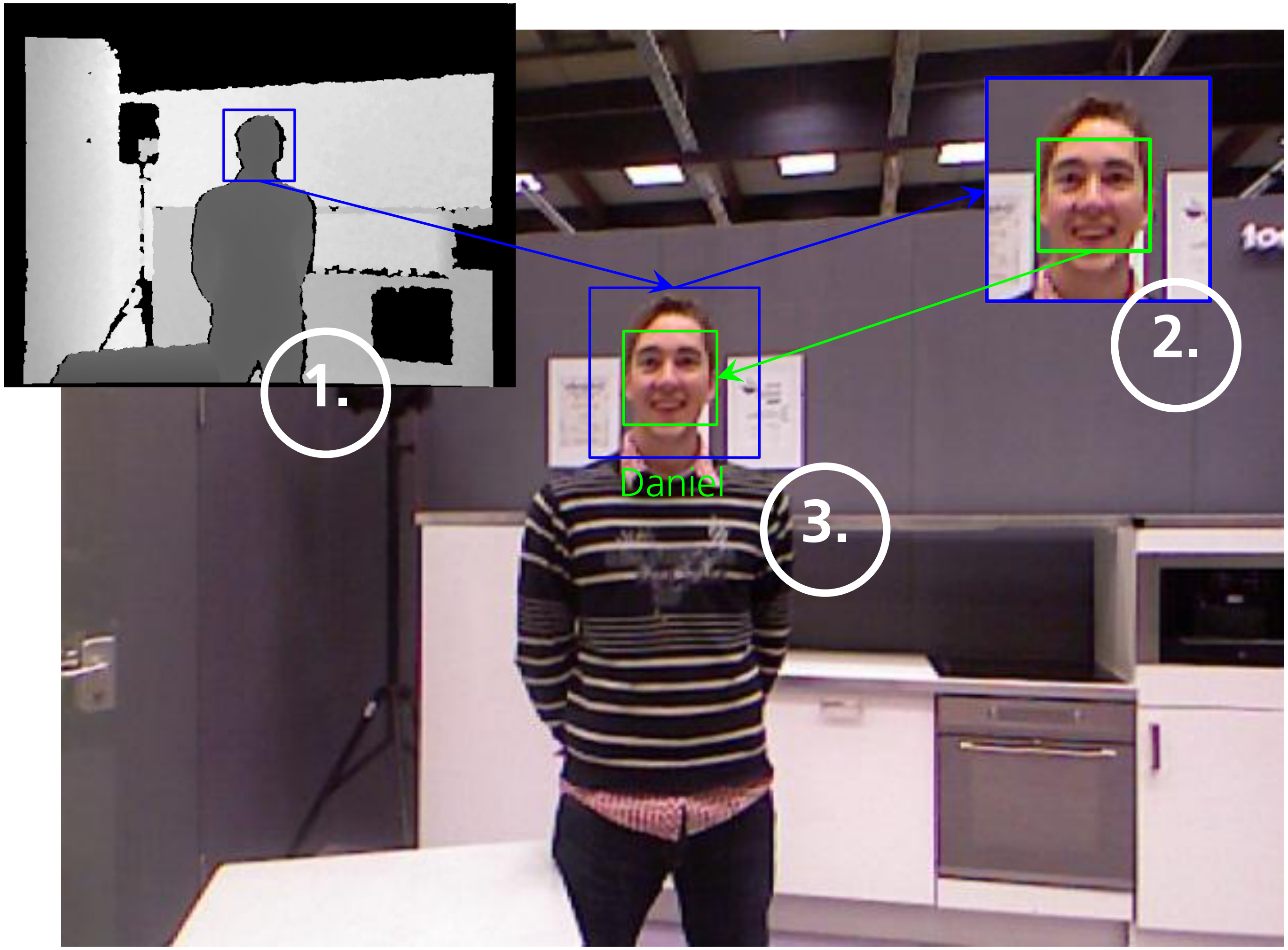


```
# Adds data to the training database
#
# goal message
string label                            # a label for the data w
int32 capture_mode                      # mode of data capture:
int32 continuous_mode_images_to_capture # if the continuous mode
float32 continuous_mode_delay           # if the continuous mode
---
# result message
---
# feedback message
int32 images_captured                   # the number of images a
```

Fraunhofer IPA

Approach for Person Detection and Identification

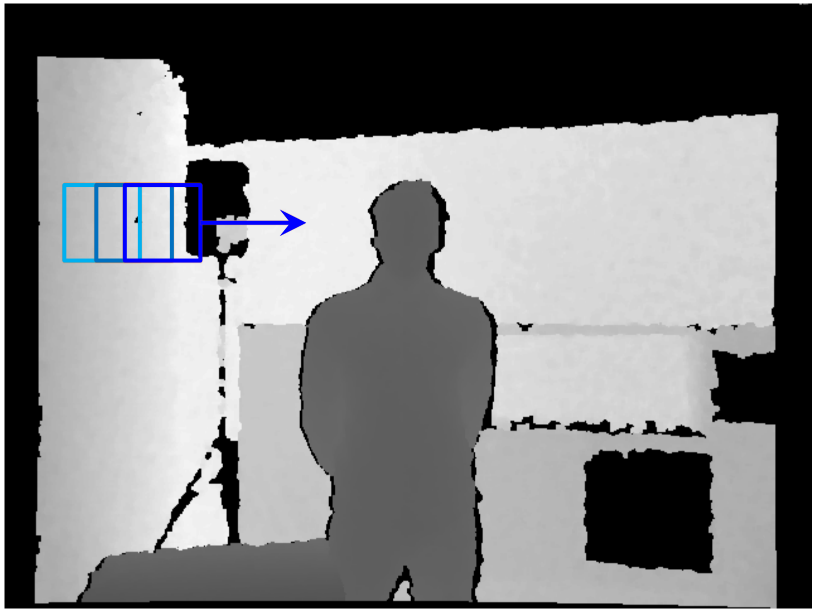
- 3 steps: 1. head detection, 2. face detection and 3. face identification

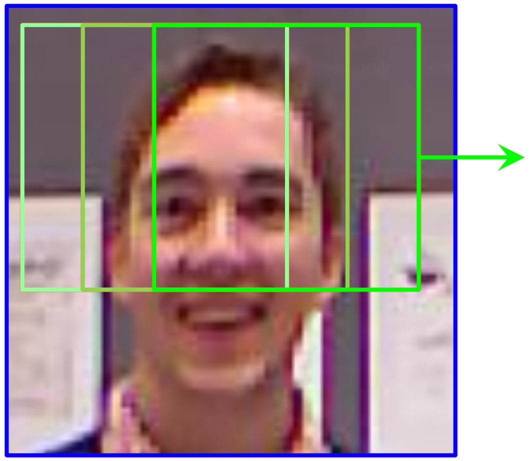


Approach for Person Detection

- head detection in depth image of the scene with Viola-Jones classifier
  $\rightarrow$ cob_people_perception comes with a well-trained classifier cascade for depth data from the Kinect

- face detection in color image patches of the head regions with Viola-Jones classifier
  $\rightarrow$ cob_people_perception uses trained OpenCV classifier



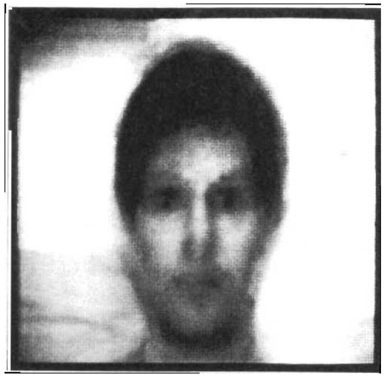


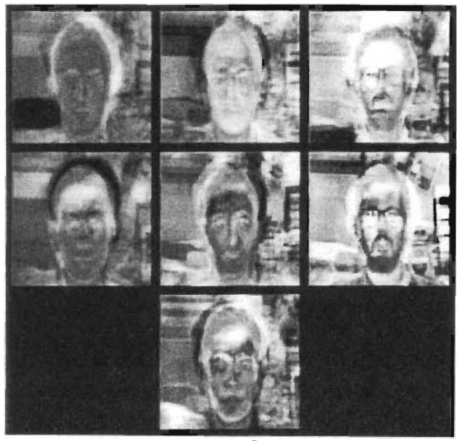
Approach for Person Identification

- Fisherfaces on gray image of the face
  - basically a projection-based method
  - generates the basis of a “face space” from training samples
  - minimizes intra-class variance and maximizes inter-class variance
  - face similarity judged by similarity of face space vectors
- intuition from similar method Eigenfaces:

known people = average image + weighted sum of eigenfaces

average image





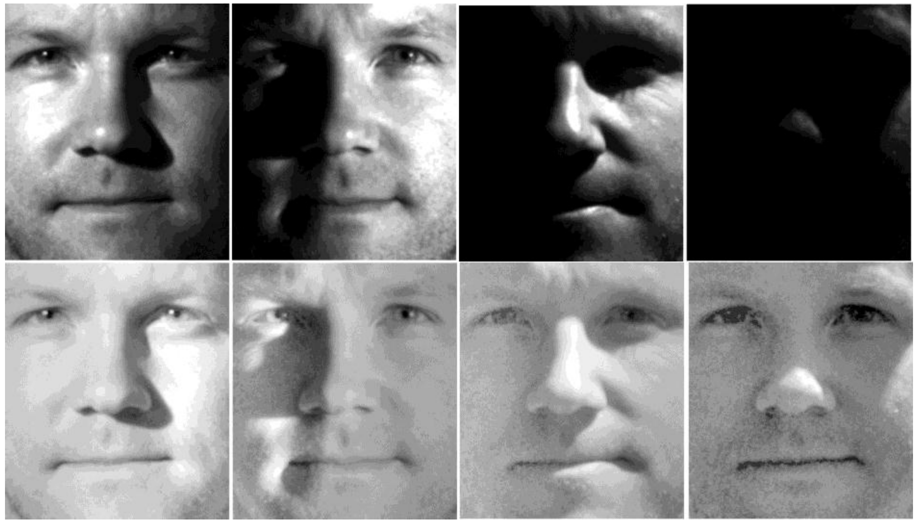
eigenfaces (face space)

images from: Matthew Turk and Alex Pentland

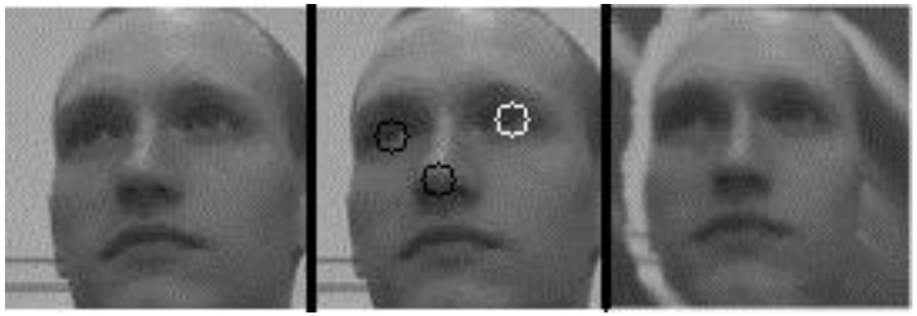
Journal of Cognitive Neuroscience 1991 3:1, 71-86

Approach for Person Identification

- increased robustness of identification through
  - illumination normalization, e.g. by
    - gamma transform combined with discrete cosine transform coefficient scaling

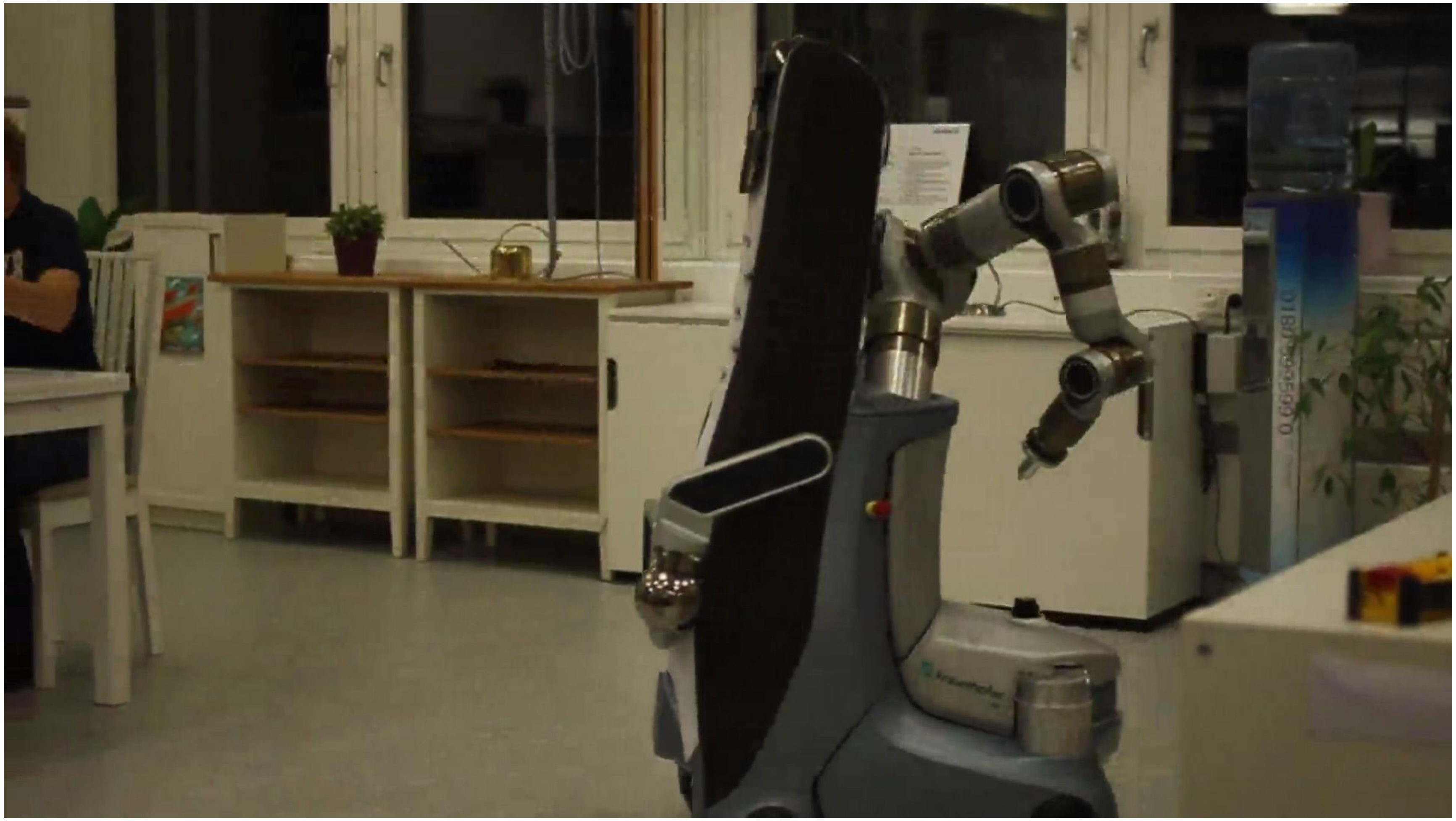


  - head pose alignment, e.g. by
    - detection of eyes and nose and image warping



Fraunhofer IPA

Video – Person Detection and Identification

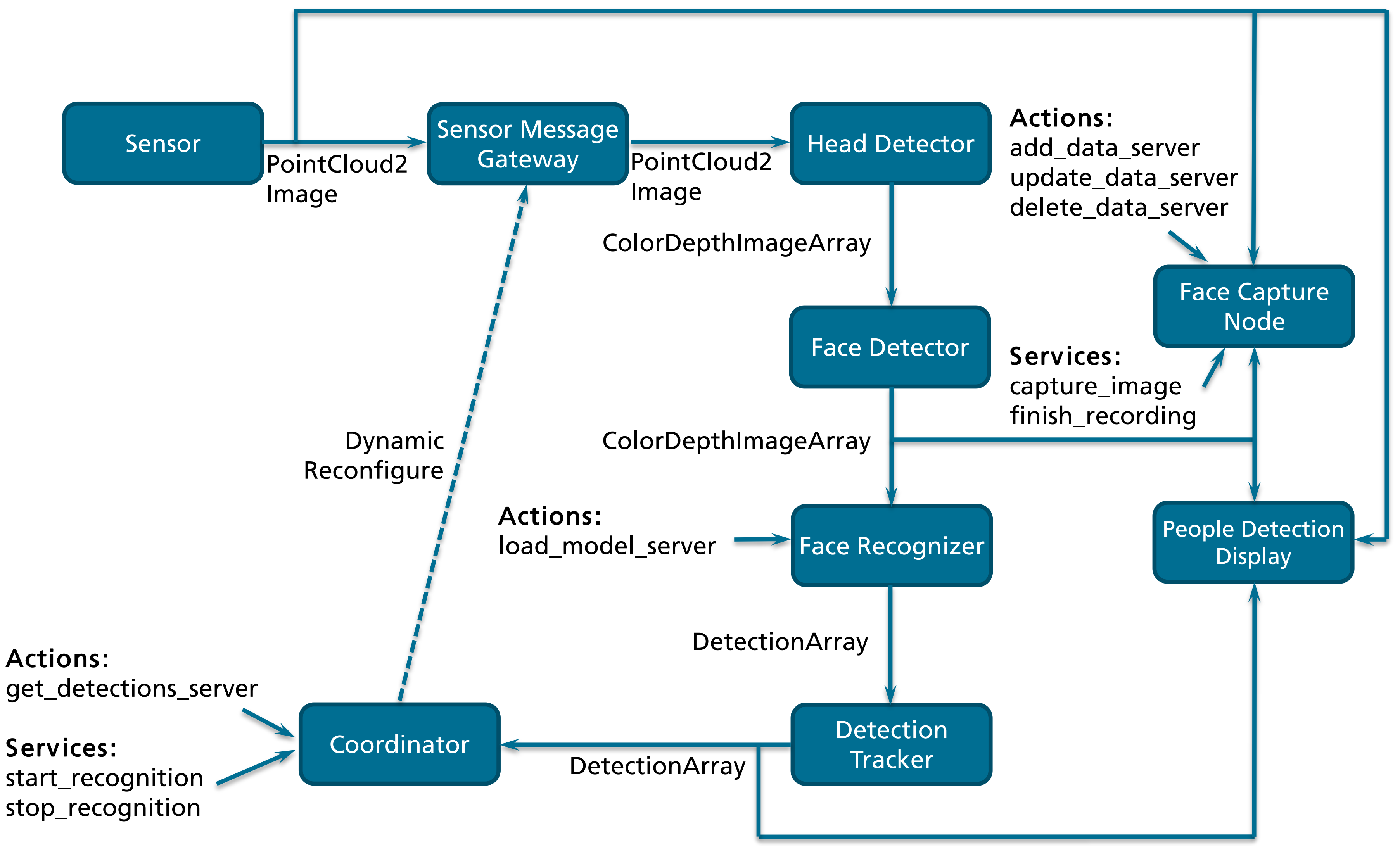


Fraunhofer IPA

Implementation

- functionality
  - person detection and identification continuously or on demand
  - capture training data for face identification
  - build face recognition models for all known people or just a subset
  - update or delete data for face identification
- division into specialized modules that run as nodes
- advantages and drawbacks of high modularity
  - + exchangeable algorithms
  - + stable interfaces
  - + uncomplicated team work on different nodes
  - - message transfer overhead

Implementation



Fraunhofer
IPA

Interfaces – Recognizing People

- continuous stream of detections and identifications

  - open camera message gateway with service *start_recognition*
    message type: recognitionTrigger.srv:
    ```
    # request message
    float32 target_frame_rate
    ---
    # response message
    ```

  - close camera message gateway with service *stop_recognition*
    message type: Empty.srv

Interfaces – Recognizing People

- single detection and identification on demand

  - request via action *get_detections_server* (opens message gateway automatically if closed and resets its status afterwards)
    message type: getDetections.action

```
#goal
float32 maximum_message_age
float32 timeout
---
#result
cob_people_detection_msgs/DetectionArray detections
---
#feedback
```

Interfaces – Capturing Training Images of new Persons

- **manual capturing** mode: call service *capture_image* to record an image, and *finish_recording* to quit
- **automatic capturing** mode: automatic recording of *x* images with a timeout of *y* in between
- request via action *add_data_server*
  message type: addData.action

```
# goal message
string label
int32 capture_mode         # 0=manual, 1=continuous
int32 continuous_mode_images_to_capture
float32 continuous_mode_delay
---
# result message
---
# feedback message
int32 images_captured
```

- **update labels** or **delete training data** with actions *update_data_server* and *delete_data_server*

Interfaces – Construct Recognition Model

- build a model for face recognition with Fisherfaces for a specified list of persons that have been recorded to the database before
- request via action *load_model_server*
  message type: loadModel.action

```
# goal message
string[] labels   # list of persons to be recognized
---
# result message
---
# feedback message
```

Video – Demo Application “Search for a Certain Person”

Fraunhofer IPA

Usage

- get the software from www.ros.org/wiki/cob_people_perception
- compile
- run a roscore
- start the Kinect driver

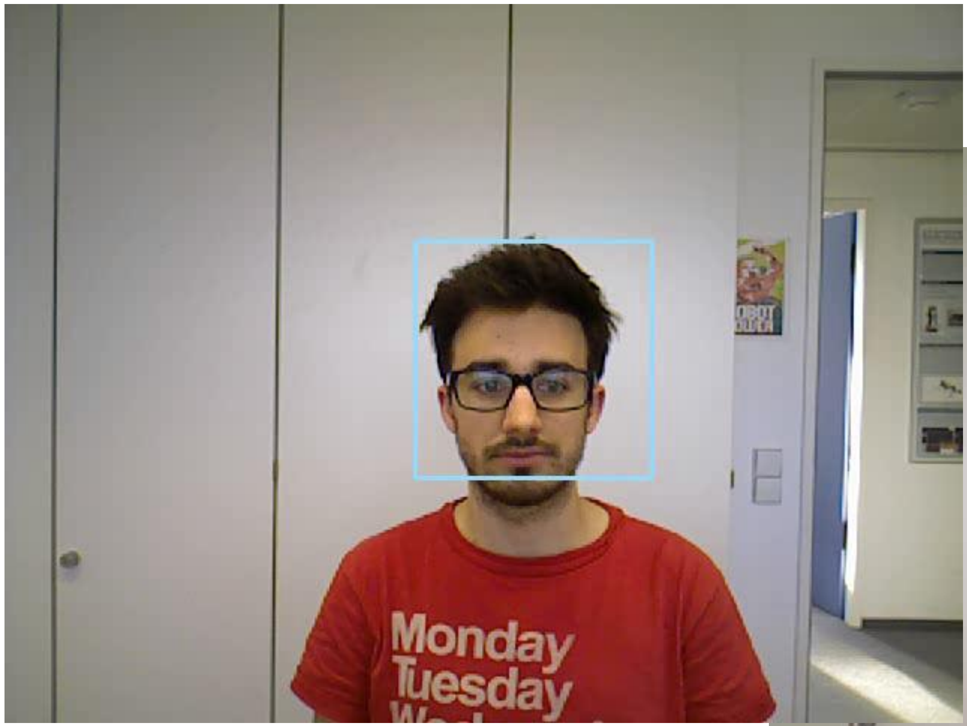
  ```
  roslaunch openni_launch openni.launch
  ```
- launch people detection

  ```
  roslaunch cob_people_detection people_detection.launch
  ```
- start the client for manual usage …

  ```
  rosrun cob_people_detection people_detection_client
  ```
- … or just start your own script or state machine to communicate with the person identification module

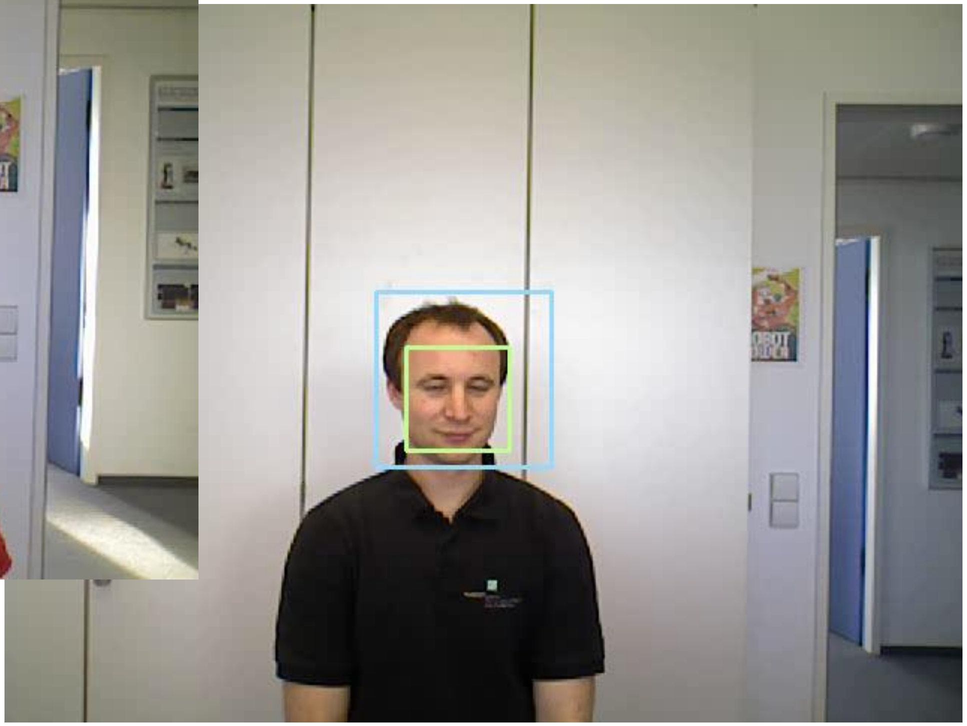
Thank you for your interest!

and thanks to the major collaborators within this project



Thomas Zwölfer

Jan Fischer



for more infos visit www.ros.org/wiki/cob_people_perception

Fraunhofer IPA