Techniques for Real-time Multi-person Face Tracking for Human-robot Dialogue

Zahra Katibeh

This thesis is presented as part of Degree of

Master of Science in Electrical Engineering

Blekinge Institute of Technology

April 2013

Blekinge Institute of Technology

School of Engineering

Department of Electrical Engineering

Examiner: Benny Lövström

Abstract:

The aim of this work is an investigation of interaction between a robot and multiple humans. The head robot is “FurHat” - in Speech, Music and Hearing (TMH) department in KTH - and the most focus is on real-time tracking algorithms. The study contains two parts: first, selecting an efficient algorithm by comparing some of the market’s offers in detecting and/or tracking area; second, establishment of a method which has the possibility of starting a real-time interaction between FurHat and multi-person. A machine learning algorithm in Matlab suggests a method to calculate the coefficients for “Priority Function” which is implemented on the outputs of SHORE library. This function can guide the robot to select the target face which should be chosen for tracking or starting a conversation. The project is done at KTH in TMH department.

Keywords: Face tracking, Priority Function, Real- time algorithm, Robot, Interaction, Multi-person.

Acknowledgements

This thesis was carried out under supervision of Dr. Jonas Beskow, at Royal Institute of Technology, Stockholm, Sweden, and Dr. Benny Lövström at Blekinge Institute of Technology, Karlskrona, Sweden.

I am grateful to my thesis examiner, Dr. Benny Lövström at BTH, for his support and guidance.

Also, I want to thank Dr. Jonas Beskow and Samer Al Moubayed , who were very knowledgeable and supportive. Their comments and supervisions help me to accomplish the thesis.

I am very grateful to my family, specially my mother for her kindness and my husband, Mehdi, for his encouragement, kindness and patience. I really could not finish the project without his support and protection.

Contents

1 Introduction

1.1 Problem statement

Over the recent years, having a good interaction between humans and robots has become one of the important subjects in many different computer vision topics and robotic communities. Since the utilizing of robots has significant increased in human life autonomous behaviour is defined to have intelligent systems for robots and lots of researches in this area.

Monitoring human behaviours is a basic method for modelling machines which want to act like human beings; however, they usually cannot support all details and circumstances. Humans has the possibility to analyse different facts and actions at the same time and most of the time reactions happens automatically. Meantime, they are learning how react and respond to the events during a long time. In contrast, in order to have any type of interaction with robots, they should be learned every things with details from the beginning steps. Needless to say that action and reaction from the machine will just be possible by writing codes and using learning methods (Machine Learning).

Many types of robots are used these days and the main competition between different algorithms and robots are based on comparing the accuracy and implementation time. Indeed, real-time responses will aid human to interact with robots naturally and, at the same time, increase the application of using machine in everyday life.

In this context, we are going to distinguish an optimized algorithm for a head robot, FurHat, a designed robot in Speech, Music and Hearing (TMH) department in KTH (Royal Institute of Technology), in order to have a real-time multi-person tracking interaction with humans. One of the defining tasks for this robot is playing a secretary role in the building's entrance.

The main aspect of the proposed method is working in real-time. In fact, he might be useless if he cannot have a real-time act and react with human. So, the first factor that is considered in all investigated methods in this survey is being real-time. Moreover, tracking more than one person without any limitation in the number of trackers is the other condition which should be

noticed. As a consequence, the result of first part of this project is discovering a method with real-time implementation and multi-person tracking ability. So, the first part of the research is covered three different methods of object tracking to detect which one is more match and suitable with FurHat situation (with emphasis on face tracking algorithms). It should be noted that the method is found out from the market offers and it has a commercial license from "Fraunhofer". SHORE™ (Sophisticated High-speed Object Recognition Engine) is one of state-of-the-art real-time multiple face tracking systems which is produced in this company and the company has accepted to deliver the library of software to KTH after correspondence about this production. Needless to say, it is restricted for academic use. The next step is performing some adjustments on recognized method in order to make it compatible with the robot's demands. Before this stage, being familiar with this software and its abilities is mandatory and is required for other steps.

Studies on this product lead the project toward finding a method which can guide the robot performance more intelligent. Hence, the next step is investigating a technique for defining a function that is supposed to detect the face with highest priority in every frame in live video. The function is specified by its parameters and coefficients. As this function is widely scattered, parameters and coefficients would be different according several situations. In this project, we are going to determine a method which can support current requirements; however it has possibility to extend more for future needs.

Figure 1.1. Head robot FurHat in TMH department.

1.2 Background

1.2.1 Face Detection:

Since object detecting and tracking play an important role in many computer vision aspects, wide range of researching has been done in this domain. There are lots of applications in image processing fields supporting with object detection. Face detection has been one of the proposed subjects for a long time and first attempts to do it began in 1960's with a semi-automated system. Recently, it becomes pervasive in different forms, for example, most of digital cameras can detect objects and specially faces before taking a photo. Also, this is very useful in video surveillance or human-computer interaction. As object detection and specifically face detection is one of the communication ways between computer and human, it is more considered in image processing these days.

1.2.2 Head Robot: FurHat

As mentioned before, FurHat is a head robot in KTH which is, currently, using advanced camera to handle a dialogue between two persons. Using advanced cameras, especially Kinect camera, is remarkably useful in robotics field because it has some great facilities which are available with reasonable prices. Despite its limitation, it is a real-time tracker and it can be utilized in human robots interactions. However, its limitation will not allow the robots track more than two person at the same time. Moreover, the distance between the robot and tracked person depends on its sensor which is just valid in the specific distance (1.2 – 3.5 meters) and it would be not enough accurate outside of this range.

1.3 Objective

The main purpose is to investigate through different methods and figure out a way in order to have a real-time multiple-person tracking algorithm for FurHat. The first step is based on comparing some kind of available methods such as using advanced cameras, colour tracking and/or face tracking with emphasis on commercial licenses or open-source research

prototypes face tracking because this thesis is not directed towards building a state-of-the-art face tracking systems.

The second part is focused on presenting an algorithm to recognize the specific face in a live video (in every frame) which could guide the robot to detect that face. That specific face is the face which is more interested to start a conversation with the robot according to some parameters. This would be possible by providing an algorithm that is based on a priority function, with parameters and coefficient. This function should have the possibility of selecting one face in every frame, among all existent faces. For example, five faces are detected in figure 1.2 and the robot is going to select only one face to look at or maybe wants to starts a conversation. So, the priority function would choose one of these faces according to the defined parameters.

Figure 1.2. Detected faces by the SHORE.

1.4 Scope

Due to the limited time of this project and as the face detection is an extensive area in computer vision and especially in human robot interaction, all available or methods cannot be considered in this research. Therefore, the main focus is first on figuring out the best algorithm - among those are analysed during the beginning steps of this project - and then gain a way to have a communication with the robot by producing an output which can be as an input for the robot.

From the results point of view, the focus is on finding a real-time multi-person algorithm with less limitation or strict in number of trackers or distance with the camera. In the part II, the aim is to establish a method based on machine learning in order to detect the specific face in each frame by defining a priority function and some parameters.

Many different types of object detection methods are available but the most emphasis is on face tracking algorithms rather than colour tracking or advanced cameras. It is not possible to investigate all of them in this report. So in this context 6 different methods (among all of those that are checked) are considered for real-time face tracking.

In addition, the definition of priority function is not unique and could depend on lots of different parameters and, also, the method of finding the coefficient can modify by using advanced machine learning algorithms; however, the results can cover the requirements of this project in this step.

2 Methodology

In near future, robots are going to have an impressive role in human's life. Their usages would be contained a wide range of variety and one of their common aspects would be interacting with human beings. So, in order to have a humanoid interaction, robots should have the ability of detecting, tracking and responding to objects or people. Therefore, a robust accurate algorithm which has the possibility of implementing in short time is required.

Real time algorithm is the first step for having a real interaction, that's why the researchers try to use less complicated algorithms and they would accept that these algorithms may have some disadvantages. They also should consider the result accuracy, as these days the robots are used in many professional different aspects. Combination of being real time and implementing accurate could be the desired result for lots of research areas.

2.1 Current situation

Different patterns of utilization could be defined for a robot. In this context the focus will be on tracking the object and specifically the face. Indeed, a real time object tracking algorithm is desired for the existed robot - FurHat, a designed robot in Speech, Music and Hearing department in KTH - which is a human-like robot head for multiparty human-robot interaction. The robot should have the ability of starting, controlling and continuing a conversation with multiple humans. Currently, he can take part in a conversation by using the facilities of Kinect camera and tracking the person who is talking -by moving his head [1]. In this case, we can detect the depth in visual tracking and also recognize the speech by using an array microphone. For the sake of some restrictions of using advanced camera (such as minimum and maximum distance between the camera and person), finding a comprehensive solution is mandatory. It is supposed to have the ability of tracking the gaze with his eyes and head poses in different condition like human in near future also.

The main restriction of using Kinect camera for FurHat is the number of tracked people (maximum 6 trackers, simultaneously). Also, its sensor is

just valid in the specific distance 1.2–3.5 meters. In fact, these conditions would just allow the user presents in a very limited space and of course, this is not expected for FurHat. Therefore, the aim is investigating a solution to cover all these defects in order to increase the ability of the robot to become more humanoid.

2.2 Suggested solution

According to the previous knowledge and studies in this specific case in TMH department, three suggested ideas were defined to overcome these problems.

- Face Detection
- Colour Detection
- Using Advanced Camera

The first idea is based on face detection by searching in open source software to gain a real time multi face tracking algorithm which has the ability of connecting directly to the robot system or at least with some modification. Needless to say, applying face detection algorithms let the user have a more intelligent robot by detecting some extra factors such as age, gender, emotion and etc. Most serious disadvantages of using face detection algorithms could be run time and reliability which could be gradually improved by modifying.

The second one is tracking the colour instead of the face in order to have a real time algorithm. It can be useful if it supports the main idea by simple and reliable solution. Colour detection will be done by using a coloured sticker for every object which would be tracked. The robot should detect and track the specific colour as long as that colour is in front of it. As a matter of fact, increasing the probability of errors and difficulties of using coloured sticker are disadvantages of this method.

The third offer confirms to improve the usage of advanced camera by integrating it with another method (for example by tracking colour at the same time). Although, Kinect camera is a motion sensing input device that

is quite possible to track a person by using it, it has some limitation in the number of trackers which does not let applicators using it in general.

First step of investigation of this task is going to clear some basics about face detection and tracking for robots. This part considers some of the researches which have been done in human-robot interaction area. Then, we are talking about three suggested methods by explaining more details and with emphasis on face detection algorithms and available market offers. Next step is comparing these three techniques and presenting the advantages and disadvantages of each algorithm in order to select the best one for further steps.

Therefore, some initial information about every method will be presented in order to select an appropriate existing one (or develop a new method by using open source algorithms) and then the result could be concentrated on the best solution. The last part is about the chosen method and the way that is used for our purpose.

3 Face detection

Human robot interaction has been an active area for a long time and has been widely improved recently. As a pre step, it is required for some other algorithms such as face recognition, facial expression recognition, video surveillance system, gender classification, facial future extraction, face tracking, clustering, attentive user interfaces, digital cosmetics, biometric systems, digital cameras and specifically robot vision interaction. Face detection also contains lots of application areas like clustering, tracking and picture indexing [2] [3]. It is possible to use any kind of algorithms based on face detection for human-robot interaction, however, sometimes they cannot cover all the required conditions such as accuracy and low error rate.

The face detection's procedure can be divided as:

- Pre-processing step
- Face detection step
- Post processing step

The first step is including the image processing techniques, the second step could be defined in different algorithms of face detection and for the post processing step the method could support to detect the faces properties (facial size, location, rotation…) which are at the same positions to have an acceptable false rate error [2].

Although face is a main source of information during this interaction, detecting a face in an image or movie has some challenging difficulties. The main problem for detecting the correct face will appear because of wide range of variety in facial shapes and scales, facial features, illumination and occlusions.

For instance, using pattern classification approaches is very popular in most of face detectors. In this case face can be recognized from the background by a classifier and non face patches is trained from a set of training examples [6].

Also, for implementing the face detection, various pre-processing on the images is essential; for instance, image should be processed in different aspect to detect the edges and eyes or using different tools like histograms or graphs in order to detect the face in every single picture. In other words, for reducing the illumination effect, histogram equalization or standardization is necessary.

3.1 Face detection algorithms

Face detection has become a demanded technology for years and it has achieved much remarkable progress till now. One of the most famous face detection algorithms is Viola Jones face detector which has had the most impact in the 2000's; it would be most likely the seminal work by Viola and Jones. The Viola-Jones face detector contains three main ideas that make it possible to build a successful face detector which has the ability to run in real time: the integral image, classifier learning with AdaBoost, and the attentional cascade structure [4].

In fact, face detection area contains a wide range of pre-study knowledge which can be classified in different methods. Yang et al. [7] grouped the various methods into four categories [4] [5]. Each method might categorize in more than one group as they have some overlap.

- ***Knowledge based method***
  It is usually based on rule, especially human coded rule, in order to encode the information of human face such as eyes or/and mouth.
- ***Feature invariant method***
  It includes structural features which do not depend to pose, lighting and rotation of the face. For example, skin colour.
- ***Template matching method***
  This method is based on comparing the input images with the preselected images to find correlation between them.
- ***Appearance based method***
  Appearance based method is similar to template matching method, however, the models are learned from pre labelled training set and one of the essential algorithms is eigenface method in this category. Neural networks, Bayes classifiers, support vector machine, AdaBoost based face detection, Fisher linear discriminant, statistical classifiers and hidden Markov models are all in appearance based methods.

3.2 Evaluating face detection algorithm

These following terminologies could evaluate the effect of the performance of the method.

- ***Detection rate***
  The proportion of faces which is detected correctly by the algorithm (in each training set) to the exact number of faces which is exist in that set is called “detection rate”.
- ***False positive***
  The number of objects that has been detected in the pictures as a face wrongly is “false positive”
- ***False negative***
  The number of faces that exist in the pictures but the algorithm could not detect them as a face is “false negative”.
- ***False detection***
  As it is clear from its name, the total number of false negative and false positive in every training set is “false detection”.

3.3 Face detection and tracking in human-robot interaction

One of the requirements for human-robot interaction is preparing a natural conversation between robots and surrounding people which would be available by diagnosing and tracking the face. Indeed, for the next generation robots which should have the ability of utilizing in domestic, entertainment market or service area, Human Robot Interaction is a key technology [11]. Since the demand in this area is increasing gradually, lots of different usage might be defined for the robots. Utilization of the robots would not be limited in some cases but at least some of them could be listed in the following context.

Case A: H. Kim et al. develop a technique which is possible to localize the speaker by utilizing audiovisual integration. They propose an interaction system between humans and robots by using speech signal and face

tracking simultaneously and deploy it in a prototype robot, called IROBAA. In particular, for having a natural interaction the robot first estimate the distance of speech signal then find the direction and finally turning its head to the direction of the speaker's face. In this stage, robot involve in detecting a specific face which might be tracked for further interaction. The proposed method has expected to decrease the error of localization of the speakers because by integrating visual and auditory processing technology the errors in the speaker localization could be compensated by visual information. Needless to say that finding a speaker by just using auditory system is more likely to fail due to environment's noise. IROBAA can locate the speaker at whole direction (0-360) using three microphones and it needs a computer for calculating [8].

Figure 3.1. IROBAA (Intelligent ROBot for ActiveAudition)[8].

Case B: C. Tsai et al. develop a face tracking interaction control system for a wheeled mobile robot by using visual tracking control techniques. The proposed task for this robot is controlling the motion, detecting the target, estimating the depth, etc. Since one of the tasks is target detection, and interaction between robot and people is one of the targets, recognition, identification and tracking are essential factors. In this case the camera (robot) and the target both are moving with different dynamics. The utilization of this robot would be in several areas such as visual navigation,

visual surveillance and etc. Designers apply adaptive skin colour search method for solving the colour changing problems –in different light conditions and they also use Kalman filter and an echo state network-based (ESN-based) self-tuning algorithm simultaneously for increasing the robustness against coloured observation noise in it. It means the algorithm has the ability of improving against not only white noise, but also coloured noise without any modification [9].

Case C: An intelligent face tracking implementation has been done on Lego Mindstorm NXT® Robot platform by V. Nikhra et al. which are integrated Viola Jones face detector, eye detector and the Camshift algorithm. "Camshift algorithm relies on back projected probabilities and it can fail to track the object due to appearance change caused by background, camera movement and illumination. Eye detector is used as verifier while initializing the Camshift algorithm and later in face tracking."[10] Robot movement and camera movement is qualified by Camshift algorithm and it can be further used as pre-processing step in facial expression analysis, gaze tracking, lip reading or face recognition and also utilized in human-robot interaction systems [10].

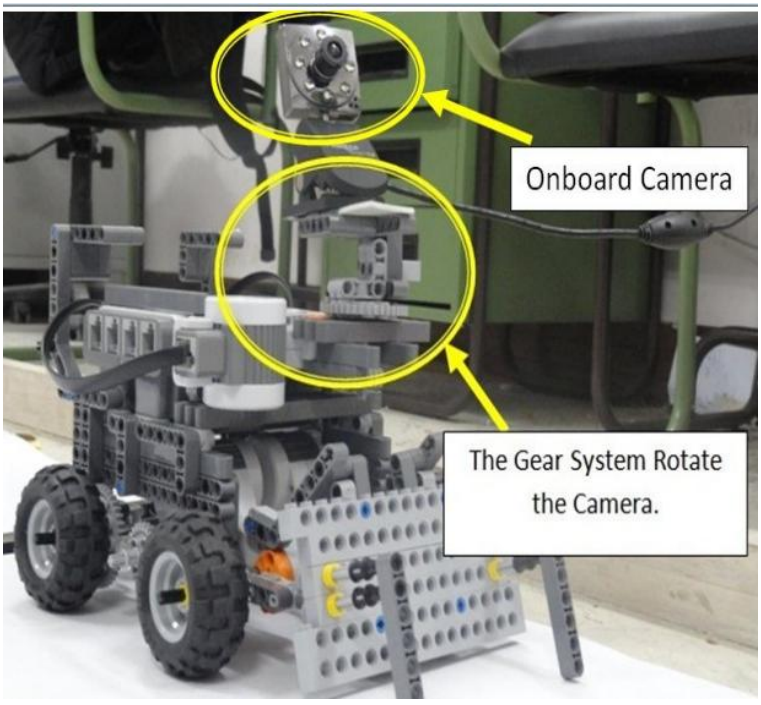
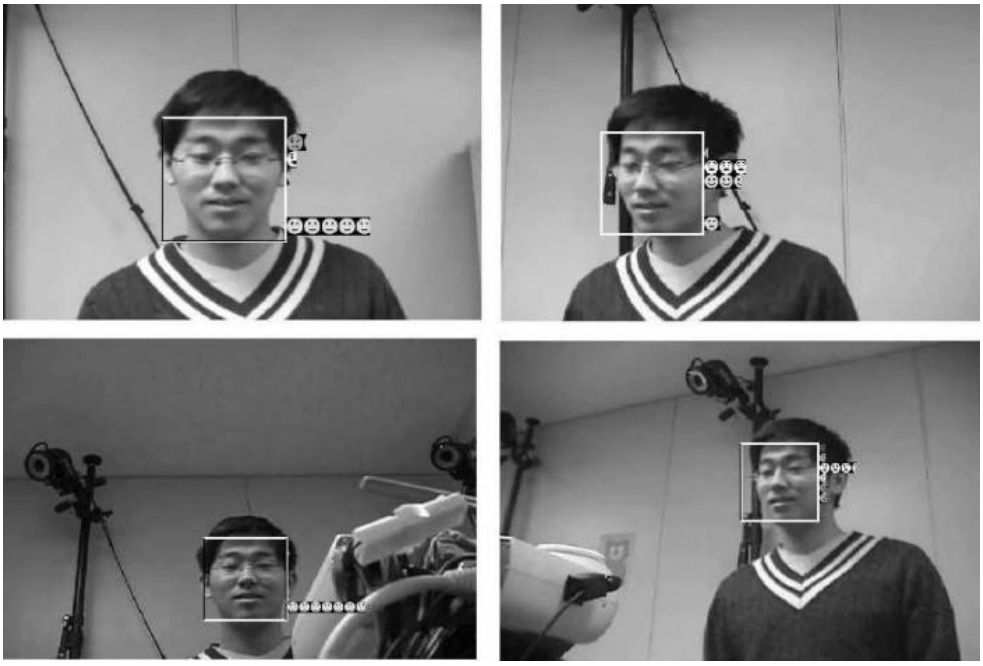


Figure 3.2. Robot fitted with camera [10].

Case D: Another researcher group, M. Pateraki et al., proposes a methodology for detection and tracking of facial features which combines appearance-based and feature-based methods for recognition and tracking, respectively. The proposed approach is intended to support natural interaction with autonomously navigating robots that guide visitors in museums and exhibition centers and, more specifically, to provide input for the analysis of facial expressions. Instead of human interaction that is generally based on their eyes, robots should usually use sensors such as sonars, infrared sensors and/or laser range scanners. One of the reasons of using sensors is its possibility of providing accurate range measurement of the environment especially in large angular field and with high rates. So their most popular usage could be in tracking and navigation robot systems. In this method, Laser Squares Matching (LSM) is used for face tracking which can model various geometric between images patch in different images. However for extra information about the face and gesture, vision is the only alternative. By integrating the data information from the laser which could find the location of the tracker in 3D, with appearance based method which is detected and tracked the face in that specified area the algorithm becomes more efficient and useful [11].

Case E: G. C. Littlewort et al. have developed a real-time face finder system which engages cascade feature detectors. It can be trained with boosting techniques in order to indicate the progress in detect frontal face in the video stream. They code the detection face in seven dimensions in real time: neutral, anger, disgust, fear, joy, sadness and surprise. The expression recognizer integrates a feature selection based on AdaBoost and a feature integration based on support vector machines (SVM) which can enhance speed and accuracy. They evaluate this system as a potential new tool for research in behavioural and clinical studies and as a method for automatic measurement of human-robot interaction [12].

(A)



(B)

Figure 3.3 (A), (B). Human response during interaction with the RoboVie robot at ATR is measured by automatic expression analysis [12].

Lots of other similar experiences are existed in this field which should be considered in next steps. In this stage we classify whole subject into following categories:

1) Colour tracking
2) Advanced camera
3) Face tracking in market survey

4 Evaluation of proposed methods

In this chapter, three proposed methods will evaluated by initial information. Being aware of weaknesses of colour tracking and advanced camera methods, will guide the evolution to concentrate more on the third method. So, here, we start with a brief explanation on first two methods and the more emphasis is on “face tracking in market survey”.

4.1 Colour tracking

Detecting and tracking coloured stickers can be used instead of using face detection. In this case a person who would be tracked by camera should use special coloured sticker. Of course, tracking colour algorithms has less complexity and need less time for run in comparison with tracking faces but it has a wide range of limitations which do not let us to accept it as a final solution.

As it mentioned, the main problem in this method is we should ask tracked person to use the sticker and it means that some catalogue should be prepare to guide them in order to use this service. In addition, selected colour should be very special so that the camera can easily recognize it from other colours in the environment (for instance clothes, bags, etc.) However tracking by colour will not allow us to get extra information about tracked persons, like emotion, gender and age, it cannot be rejected completely because of being real time and multi-track.

4.2 Advanced camera

Using advanced cameras, especially Kinect camera, is remarkably useful in robotics field because it has some great facilities, which are available with reasonable prices. Kinect can use a real time algorithm providing a 640x480 depth map that detects objects which are placed in different distances from camera.

On the other hand, disadvantage of using Kinect is that its sensor is just valid in the specific distance 1.2 – 3.5 meters and it would be not enough

accurate outside of this range. Also, Kinect will not useful in direct sunlight, however, it can work properly in darkness or night (with low light) and provides high resolution images. Another limitation of using Kinect refers to the number of tracking person simultaneously which is up to six, including two active players.

4.3 Face tracking in market survey

As the objective of this investigation is not based on building a state-of-the-art real-time multiple face tracking systems, we are going to compare some market offers.

Open source research prototypes are available in the markets and they usually have useful source codes. In chapter 4 and 5, first, some models that are based on interaction between a robot/camera and human will be presented and then by a comparison between their function, the best one would be selected. All these presented tools have been checked in the first step then some demo versions have been installed and compared to each others.

4.3.1 Detect and Recognize Faces with Luxand FaceSDK

FaceSDK identifies face and facial features with high performance and is a multi-platform function. It has the capability of recognizing facial features, detecting the matched faces on different photos and locating faces on animated video stream and images in real time. FaceSDK enables Microsoft Visual C++, C#, VB, Java and Borland Delphi developers to build Web, Windows, Linux, and Macintosh applications with face recognition and face-based biometric identification functionality. Some features of this software are: [1]

1) Compatible with 32-bit and 64-bit environment
2) Support all webcams on the markets
3) Support multi-megapixel images
4) Integrate with new or existing projects easily
5) Developers can create a wide variety of applications

[1] http://www.luxand.com/facesdk/

It also helps constructors who build complex face morphing and animation system for entertainment portals. Utilization of FaceSDK is expanded to the online and desktop solutions which require accuracy and reliability in face detection fields. Moreover, web applications, biometric login automation system, photo imaging and video processing solutions could use its library. The FaceSDK library has the following technical specifications:

Face Detection

- Robust frontal face detection
- Detection of multiple faces in a photo
- Head rotation support: -30 to 30 degrees of in-plane rotation and -30 to 30 degrees out-of-plane rotation
- Determines in-plane face rotation angle
- Detection speed: as fast as 241 frames per second*[2], depending on resolution
    - Real-time detection: 0.0041 sec (241 FPS)*, webcam resolution, -15 to 15 degrees of in-plane head rotation
    - Reliable detection: 0.267 sec*, digital-camera resolution, -30 to 30 degrees of in-plane head rotation
- Returned information for each detected face: (x,y) coordinates of face center, face width and rotation angle
- Easy configuration of face detection parameters

Face Matching

- Matching of two faces at given FAR (False Acceptance Rate) and FRR (False Rejection Rate)
- Enrolment time: 0.02 seconds (50 FPS)* (at webcam resolution)
- Template Size: 16 kb
- Matching speed: 49700 faces per second*
- Returned information: facial similarity level

Facial Feature Detection

- Detection of 66 facial feature points (eyes, eyebrows, mouth, nose, face contour)
- Detection time: 0.104 seconds* (not including face detection stage)

[2] *Measured on Intel Core i7 930 processor with 8 threads

- Allowed head rotation: -30 to 30 degrees of in-plane rotation, -10 to10 degrees out-of-plane rotation
- Returned information: array of 66 (x,y) coordinates of each facial feature point

Eye Centers Detection

- Detection of eye centers only, detection time: 0.0064 seconds* (not including face detection stage)
- Returned information: two (x,y) coordinates of left eye center and right eye center

Multi-Core Support

- The library uses all available processor cores when executing face detection or recognition functions, to maximize performance.

Library Size

- The size of the redistributables does not exceed 21MB for each platform.

One of the most important applications of this software is detecting facial feature accurate, reliable and in real time process. In particular, FaceSDK will return 66 facial features point coordinates such as eyes contours, lip counters, eyebrows, nose tip, etc after detecting the face in every picture by using sophisticated algorithm.

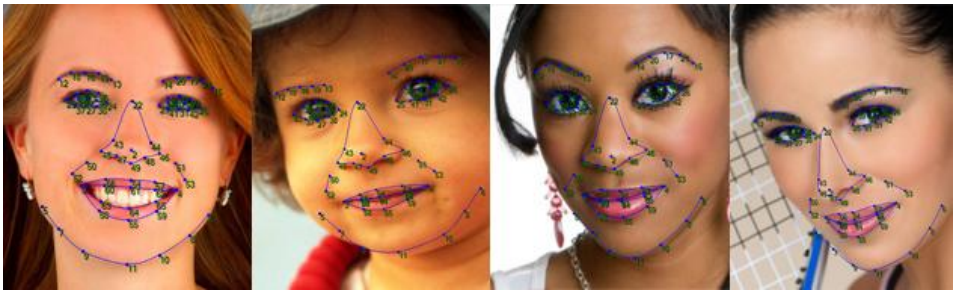


Figure 4.1. Sample of the output of Luxand FaceSDK[3]

[3] http://www.luxand.com/facesdk/

Its applications include:

- Real-time biometric authentication systems allowing the user to log in by simply looking into a webcam. FaceSDK enables touch less, non-intrusive biometric authentication.
- Automatic red-eye removal tools empowered with facial feature recognition.
- Image enhancement applications and graphic editors.
- Face animation effects for the entertainment industry.
- Graphic workflow automation systems.
- Image viewers, enhancers, and organizers with face-based image search.
- Applications for digital cameras, scanners, and webcams.
- Still image and video effects tools and plug-ins.

All in all, FaceSDK is a desire solution for detection and recognition faces, not only for developers but also for market. This algorithm is reliable and fast (real-time) and has the competence to recognize the face in different lighting conditions. Moreover, it supports all webcams and multi-megapixel images both with video streams and pictures [13].

4.3.2 myAudience-Measure

Rhohanda Software Company has developed a product which can count the audience in front of the camera by detecting the face and facial features. The production, "myAudience", is an indoor hardware-based system and would require an USB camera, a PC and permanent internet connection in order to synchronize data with central report data and update license status.[4] The application of myAudience is counting the number of people who are in the cover area and record some parameter such as gender, age and recognize if they are looking at the camera. The minimum and maximum size of cover area could be defined by the camera's lenses, model and setting. One of the aspects of utilizing this software is to analyze the reflection of the customers/audience to a specific advertisement. Moreover, assessing the images is a real-time process without recording the stream or images and the output will be prepared as matrices with several

[4] http://www.myaudience.com/

measurement parameters. The basic measurements for metrics provided number of visitors/viewers, dwell time, attention time, viewers gender/age category, time scale for each measurement. This software feature list includes:

- People counting
- Attention detection
- Gender and Age
- Reports portal
- Automatic updates and Remote troubleshooting

The accuracy of all myAudience video analytics methods is around 90% and as it works as a real time algorithm it might be one of the choices for human robot interaction with some modifications [14].

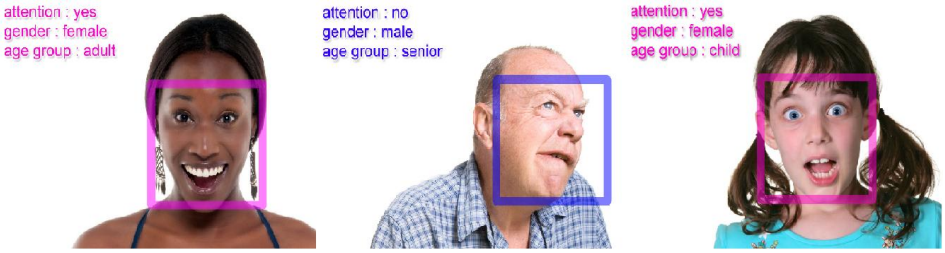


Figure 4.2. myAudience-Measure software's output [14]

4.3.3 Neurotechnology products[5]

Neurotechnology products provide some algorithm and software which is based on image processing and pattern recognition and could be beneficial in face tracking or face detection techniques. The most important products in this area are VeriLook Surveillance SDK, VeriLook SD and MegaMatcher SDK that would be explained in this context. These products could be utilized in security companies, system integration and hardware manufacturers (such as biometric finger print, face, iris, palm print recognition). The beneficial points of products are the combination of fast algorithms and high reliability simultaneously; hence, hardware providers integrate these algorithms into their products. Furthermore,

[5] http://www.neurotechnology.com/

Neurotechnology's products are used for both civil and forensic applications, including border crossings, criminal investigations, systems for voter registration, verification and duplication checking, passport issuance and other national-scale projects.

1. VeriLook SDK[6]

VeriLook is a technique for biometric systems developers and integrators and it is designed to identify faces especially in PC or Web applications. This technology assures system performance and reliability with live face detection, simultaneous multiple face recognition and fast face matching in 1-to-1 and 1-to-many modes. This product is available as a software development kit that is possible to use for Microsoft Windows, Linux and Mac OS X platforms.

Simultaneous multiple face processing in live video, compatibility with different webcams (even low cost camera), possibility of modifying for other software such as surveillance systems and supporting multiple programming languages are some of the advantages of VeriLook software. Moreover, VeriLook has a main role in advance face localization, matching face and enrolment (picture).

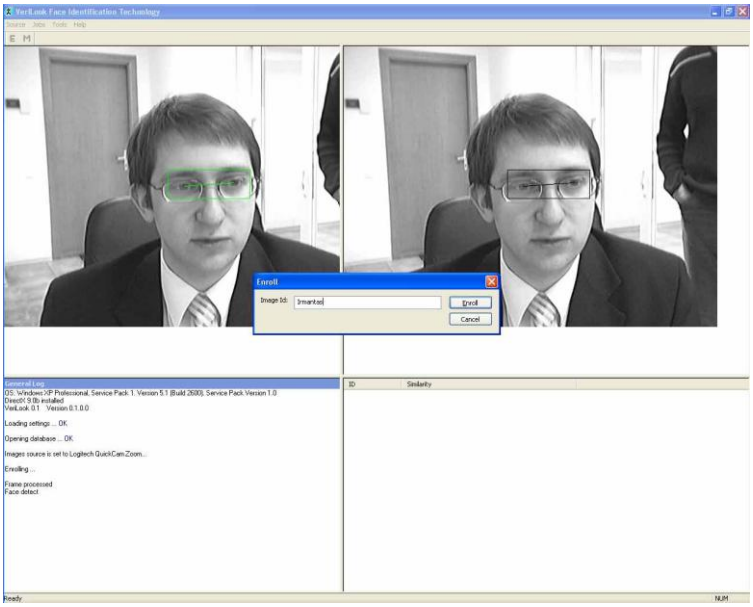


Figure 4.3. Enrol and match a face [15]

[6] http://www.neurotechnology.com/verilook.html

Its features and capabilities are [15]:

- **Simultaneous multiple face processing**
  Accurate detection of multiple faces in live video streams and still images including facial rotation (roll and yaw) and some optional feature points (both eyes, nose tip and lips middle point)
- **Face image quality determination**
  Utilizing a quality threshold to detect and save the highest quality template into database.
- **Tolerance to face position**
  Head rotating is considered. Roll: 360 degrees of; Pitch: 15 degrees in each direction; Yaw: 45 degrees in each direction
- **Multiple samples of the same face**
  Improving the biometric template record by adding different samples for one specific face in different position and time (for example with eyeglasses or without; with beard or moustache or without)
- **Identification capability**
  VeriLook functions can be used in 1-to-1 matching (verification), as well as 1-to many modes (identification).
- **Fast face matching**
  Face template matching algorithm can compare up to 420,000 faces per second.
- **Small face features template**
  Increase matching reliability by using small size for template and consequently large face database.
- **Features generalization mode**
  The face recognition quality increases by collecting the generalized face features from several images of the same subject in a database.

2. VeriLook Surveillance SDK7

VeriLook Surveillance SDK performs a real-time face identification in live video streams from digital surveillance cameras which is included face detection, enrolment, identification and tracking on video or file. The SDK is based on VeriLook facial recognition technology and is used for passive biometric identification - when passers-by do not make any efforts to be recognized. It is possible to use it in different aspects, for instance law

[7] http://www.neurotechnology.com/verilook-surveillance.html

enforcement, security, attendance control, visitor counting and other commercial applications [16]. In the picture the algorithm can detect almost all faces in the cover area.

Figure 4.4. Detecting the face in the cover area [16]

- Real time face detection, template extraction and matching against watch list database.
- Simultaneous multiple face tracking in live video.
- Automatic operation allows to log and report face appearance, match and disappearance events, as well as to enrol new faces from video stream and add them to watch list automatically.
- Large surveillance systems support by connecting several cameras to a computer and quick synchronization between networked computers.
- Available as multiplatform SDK that supports multiple programming languages.
- Reasonable prices, flexible licensing and free customer support.

VeriLook Surveillance technology extends the VeriLook face recognition algorithm for working with surveillance cameras. It has these specific capabilities:

- **Real time performance**
  Performs face detection, features extraction and template matching with the internal database in real time. The technology is designed to run on multi-core processors to achieve fast performance.
- **Multiple face tracking**
  Use dynamic faces and motion prediction models to compensate tracking the occlusion face.
- **Automatic operation**
  New faces are recognized immediately by comparing the face with database and system database will be updated automatically based on every new faces.
- **Large surveillance systems support**
  VeriLook Surveillance software has the compatibility of using multiple cameras by integrating with other surveillance systems and it can quickly synchronize biometric and surveillance data between all networks.
- **Video files processing**
  Video file is accepted as input data to process.

3. MegaMatcher SDK8

MegaMatcher is a multi-biometric identification technology especially for large-scale AFIS with high reliability and speed. It includes fingerprint, iris, face, voice and palm print modalities and can be used in national-scale projects, including passport issuance and voter deduplication. MegaMatcher supports different biometric standards and it is compatible with parallel process.

MegaMatcher includes the facilities of VeriLook Surveillance SDK - fingerprint, facial, speaker, iris and palm print recognition engines – with considering fast and reliable identification in large-scale systems. The biometric software engines contain many proprietary algorithmic solutions that are especially useful for large-scale identification problems. These solutions were specifically developed for MegaMatcher, incorporating aspects of the VeriFinger, VeriLook, VeriSpeak and VeriEye algorithms [17].

[8] http://www.neurotechnology.com/megamatcher.html

4.3.4 FaceL: Facile Face Labelling[9]

FaceL is a simple training tool to label the face and emotions in a live video. FaceL's face detection algorithm is based on open-source software and it has written in Python using wxPython as a GUI framework. Also, FaceL has another parts includes computer vision and machine learning algorithms. Combination of all these methods has prepared a tool which has the possibility of detecting and labelling the face. FaceL contains three steps:

- Detecting the faces has been done by using openCV Cascade face detector and it based on Viola and Jones "Robust real-time face detection" and Lienhart and Maydt "An extended set of haar-like features for rapid object detection" which are real-time and accurate. The process contains scanning images in the input and the output is a set of rectangles with face locations.
- Eye locations are detected in FaceL with an ASEF filter and it helps the algorithm to scale and align the face in order to decrease image variation.
- FaceL uses simple Support Vector Machine (SVM) to recognize people however the number of detected face is small. Moreover, SVM should be trained for enrolling and labelling the face. Labelling is made faster by using an eigenvalue decomposition of the training images

[9] http://pyvision.sourceforge.net/facel.

Figure 4.5. FaceL User Interface. (Right) Clicking in a face detection rectangle enrols the face. (Left) After training Foreheads are labelled[10].

FaceL can label any description of the face, emotion and position such as left and right, happy and sad and name of the person [18].

4.3.5 faceAPI[11]

faceAPI is a robust algorithm for detecting and tracking faces and it can deliver faces information such as position, rotation, facial expression and features even if they move. “These powerful image-processing and face tracking modules are combined into a complete API toolkit that other products or services can incorporate with the output.” [19] The input can be any kind of camera or webcam that will be changed into a 3D face tracking device directly by faceAPI. The main advantage of faceAPI is simple integrated with other products and application to support tracking and identifying the face. Moreover, they claim that it is the only commercially supported toolkit for developers leverage real-time face-tracking [19].

[10] http://pyvision.sourceforge.net/facel.
[11] http://www.seeingmachines.com/product/faceapi/

Figure 4.6.Sample of face API outpu[12]t.

Furthermore, "faceAPI tracks the position and rotation of the head in X, Y and Z, relative to the camera. Know where the users are, know if they're paying attention, and detect head nods, shakes and other head gestures. This six degree-of-freedom tracking information is normally only available with specialized motion capture hardware." [19]

Real- time performance, robust algorithm, automatic 3D face tracking, row image memory interface and tracking in live video are the main features of faceAPI. Some others are:

- Tracks 3 points on each eyebrow and 8 points around the lips.
- Pose-normalized face texture output, annotated with facial feature coordinates.
- Works with any webcam.
- Can track up to +/- 90 degrees of head rotation;
- Robust to occlusions, fast movements, large head rotations, lighting, facial deformation, skin colour, beards, glasses etc.
- Full control over all tracking parameters for purposes of software integration.
- Works in visible and IR lighting.

These days, lots of applications are defined for face detectors or face trackers algorithm. In the same way, faceAPI - because of some specific features- can have more application such as human robot interaction, Artificial Intelligence (AI) interaction, smart screens, interactive 3D games, intelligent video conferencing and 3D displays. It gives the developers a quick solution in face tracking aspects.

[12] http://www.seeingmachines.com/product/faceapi/

4.3.6 SHORE™ - Sophisticated High-speed Object Recognition Engine[13,14]

The SHORE™ software has a robust algorithm to detect unlimited number of faces in a live video (or single image) and track them as detect some other features. The advantage of using SHORE™ in comparison to other similar products is that it can recognize and estimate more features of every face. SHORE™ returns the positions of face, both eyes and mouth for every face. It also detects the gender classification with rate 94.3% on BioID dataset, in-plane rotation up to ±60˚ and out-of-plane rotation up to ±90˚. Moreover, SHORE™ can estimate age and recognize facial expression such as "Happy", "Surprised", "Angry" and "Sad".

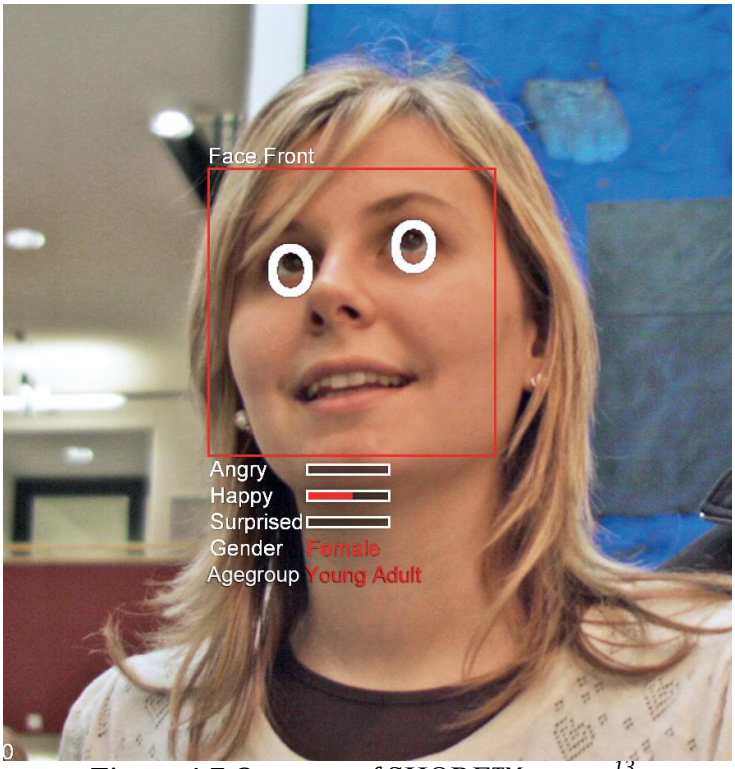


Figure 4.7.One type of SHORE™ *output*[13].

[13] http://www.iis.fraunhofer.de/en/bf/bsy/produkte/shore.html
[14] http://www.iis.fraunhofer.de/en/bf/bsy/fue/isyst.html

In addition, this software is able to detect faces down to a minimum size of 8 x 8 pixels. Output data is available for analyzing and/or summarizing. General applications for SHORE™ software are:

- Advertising and Market Research
  Measuring the attention time and considering the reaction (with correspond emotion) would be very helpful for assessing the effects of advertisement. Therefore, it could be suitable for qualitative and quantitative market research projects.
- Telepresence and Teleconferencing
  Detecting and analyzing face has the main role in conversation and SHORE™ can support these two factors on a large discussion.
- Virtual Actors and Intelligent Agents
  Non-verbal communication between humans and animated characters becomes possible because in-depth facial expression analysis can transfer a person‘s facial expressions to an animated character.

SHORE™ software solution has a very simple C++ interface and descriptive demo which helps the customers use it more easily. Also, having really fast and reliable performance, detecting gender and facial expression and estimating the age make it more useful in some other applications [20, 21]

4.4 Analysis methods

Among all these assessed methods which has been presented here, real-time ones is the desired method. The next factor should be considered is the reliability of the output results. Since all these demos are analyzed for FurHat, every other parameter should have evaluated with this robot properties for maximum compatibility. Table 4.1. demonstrates of comparing some parameters that are comparable and has more important role in the decision. For example, tracking multiple people at the same time -with minimum effect on performance in real-time- has the highest priority; also detecting the face in almost all conditions (independent from the skin color or lightening) and supporting rotated face is the other important factor. In addition, as FurHat might track a specific face for longer time,

short term recognition would be beneficial. Detecting emotion create an opportunity for FurHat to get some feedback from the conversations.

	Multiple tracking	Face rotation	Real-time*	Labelling	Lip-eye	Emotion	Gender	Age
Luxand FaceSDK	√	×	3	×	√	×	×	×
myAudience-Measure	√	√	1	×	×	×	×	√
Neurotechnology	√	×	2	×	just eye	×	×	×
FaceL	√	×	2	×	just eye	×	×	×
FaceAPI	×	√	1	×	√	×	×	×
SHORE™	√	×	1	√	√	√	√	√

Table 4.1.Comparison between 6 presented methods.

* Algorithms are sorted by their real time levels. One goes to highest rate

"×" means that the method cannot support or signify this parameter and "√" means the method has this facility.

4.5 Choosing the best approach

By considering the parameters that have direct impact on FurHat's functions, it would be obvious some of these market offers should be ignored in selection processes. For instance, faceAPI has really robust algorithm with a very high degree of reliability in face detecting area. Although, it can support occlusions, fast movements, large head rotations, lighting, facial deformation, skin colour, beards, glasses etc, it has no ability to track more than one face in each frame or image. So, faceAPI could not be one of the desired methods.

FaceL and Luxand FaceSDK can track multiple faces in live video and Facel detects the place of both eyes in every face while Luxand FaceSDK detects 66 facial features point such as eyes, mouth, nose and eyebrows.

The weakness of these two methods is running time. However they are real-time algorithms and detecting the face in almost every frame but by increasing the number of faces and/or add some extra analysis, for future application, the running time will increase too.

The Neurotechnology products demo is not enough strength in detecting the face immediately. It is really powerful method which can support lots of market applications, but the required application should be fast enough to cover the other stages such as transferring and analyzing data, also having a simple interaction system is mandatory.

The products of Rhonda Company - myAudience - and Fraunhofer Company - SHORE - have the capability of covering almost all demands for interaction between human and the robot especially for FurHat. With more focus on SHORE, it is clear that the demo version of SHORE software has some facilities to identify the gender and emotion with high reliability. Moreover, detecting face is so fast that can recognize new faces in every frame immediately. After more consideration about the details of requirements and initial correspondence with Fraunhofer Company, they accepted to provide the library of the software for academic aspects. From this stage the whole project is centralized to interact with SHORE software to satisfy the main purpose for FurHat.

4.5.1 Shore properties

As mentioned before, SHORE is a suitable connective tool for preparing a real-time multiple person interaction between humans and robots, especially for our case, FurHat. With regards to all available market survey in this field, SHORE has more competence due to some factors is described in continue.

As a matter of fact, Fraunhofer’s solution for detecting the face is based on a long period experience and the result is an object detection engine which enables to detect and analyze objects and faces. For machine learning purposes, this engine access a large amount of data with database of over 10,000 annotated faces available for training process. By this strong database and using a combination of landmark points and learning algorithms, the training models with high detection rates is accessible.

Frontal face detection	91.5% detection rate, with 10 erroneous detections	CMU+MITdataset
Gender classification	94.3% accuracy	BioID dataset
Age estimation	Mean absolute error in years: 6.85	FG-NETdataset

Table 4.2.Algorithm's performance according different datasets.

According to the table the rate of detecting frontal face and the accuracy of gender classification is above 90% which is determined the high reliability. Likewise, in order to have a real- time process, not only the algorithm but the hardware could have a direct impact on the results.

5 SHORE Algorithm Operation

Different stage of analyzing and implementation of the project will be describe in this chapter after choosing SHORE software as a conjunction tool for preparing an interaction between the designed robot - "FurHat" - and human beings. First of all, the facilities and qualifications of the selected algorithm should be evaluated in order to utilize the maximum capability. Then these qualifications would be compared with the robot's requirements for the specific usage and after solving the difficulties the suggested result will be represented.

5.1 SHORE qualification

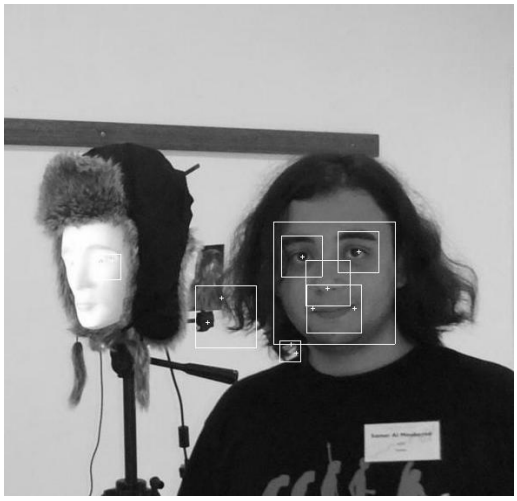
As mentioned before, SHORE is one of the best tools that we have found for this case and by assessing its competency it would be clear that SHORE has more abilities to use for future modifications. It is required to use a suitable interface to utilize and guide the current facilities for final purposes. As the main source is in C++, all other steps (except machine learning part) has been done in this environment and the focus is on running the code for both on-line camera and set of pictures. Some of the parameters which have been used in this project are listed below.

After running the main source of the program, some parameters are calculated and available as outputs [21]. Here, most of parameters that we have used during the investigation are listed. Other parameters that are provided by this software will be listed in Appendix A.

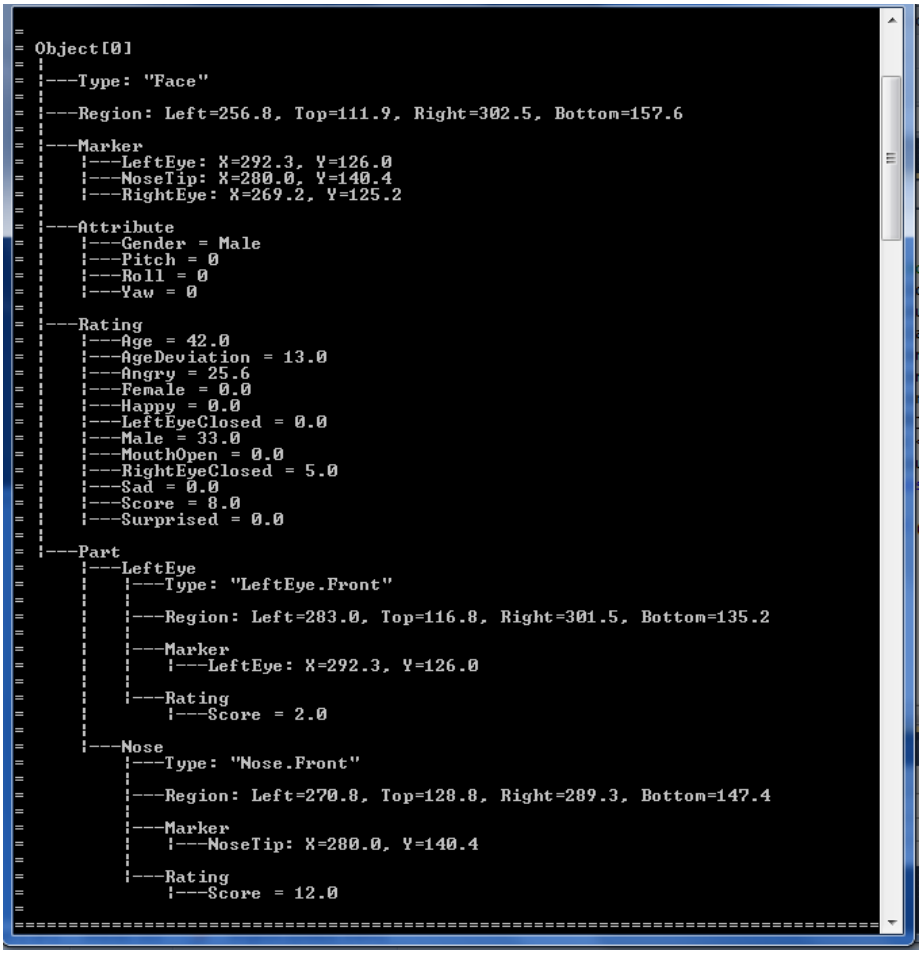
1. Coordinate:
   It contains the coordination of the face region (the coordination of left, right, top, and bottom of a square around the face) and the X and Y positions for the left and right eyes and also nose tip.
2. Emotion:
   Four types of face's emotions can be detected by this algorithm; angry, happy, sad and surprised. Each of these emotions is scaled with a number between 0 and 100.

3. Face rotations:
   The face rotation in three dimensions is presented with pitch, roll and yaw (which do not completely work in every condition in this version of software.
4. Object number:
   Every face has the specific ID which could be unchanged if the face disappears for a short time from the camera or if it becomes block with something else.
5. Gender and Age:
   Two types of outputs are available for gender; one of them shows the gender if it is “Male” or “Female” and another one returns a number between 0 and 100 for every case which would be useful in different aspects. The estimated age is also one of the outputs that has completed with another parameter “Age Deviation” to determine the approximate error.
6. Mouth, Nose and Eye:
   There is “mouthOpen” parameter that becomes 100 if the mouth is fully open. And also, two other parameters which will change if the right or/and left eyes are closed/opened. Some more parameters specify the exact position of nose and eyes with their regions.

One type of detecting the face with borders around face, eyes, mount and nose are shown in below figure. Part (B) is demonstrating some of the outputs parameters.



(A)



(B)

Figure 5.1(A), (B). Output's example and properties of the detected face.

5.2 Algorithm's requirements

In order to satisfy the requirements for having a real-time interaction with "FurHat", some of the SHORE's outputs are utilized. In the rest of this chapter the chosen parameters and their usage will be explained. The different types of parameters that are provided by this production of Fraunhofer Company is listed in Appendix A.

This head robot is supposed to be used as a part of conversation, it means he must have the ability of starting and/or following a conversation between two or more members. In case that we have just one person in front of the camera, the objective is detecting and tracking that specific face and when we have more than one face in the conversation then he should possible to switch between faces as well as detect/track the face who wants to continue that conversation or start to talk.

Generally, following some properties of real human conversations patterns will be a best guide to model a natural one for human robot interaction. For example, one of the factors that have a effective impact on choosing the other parts of the conversation is the distance. It is like a rule that people usually start to talk with a person who is closer to them if they have no other specific priority. So, measuring the distance between the robot and every other person who are in the camera's field of view could help the robot to find the nearest one. However, human mind do it automatically. Of course some extra information is needed to select the best choice. For instance, another important factor is angles. There is two kinds of angels can be defined for faces:

- The angle of the face (body) with the camera which represents if the person is standing in the corners or in front of the camera.
- The head angle with neck in three different dimensions; pitch, roll and yaw.

Angle of face with the camera (camera here is a robot head) could be defined by "X" and "Y" in a camera plane and when someone is exactly in the center of this camera plane the angels is (0,0) and in the corners of camera's field of view "X" and "Y" values become further away from the origin (0,0).

The second type of angle that is also important for find the more interested addressed is the face rotation. Evaluating this factor means a person may be very close to the other person or camera but the face has an angle with the body and they are actually not face to face. In a natural speech rotated face is rarely selected to start or continue the conversation if someone else is directly looking at us. As mentioned before, the roll, yaw and pitch angels are calculated and delivered by the SHORE's output but unfortunately we cannot have all these three at the same time in this version of SHORE software. So, as it will be explained later, the more effective parameter for our purposes in this investigation is the yaw value but just the value of

"Roll" is available with the current version of software and it is considered in the next steps.

Another parameter that we can consider in order to select an interested face in a frame to start a conversation, might be happiness. According to the software's outputs we have a parameter which is presented the emotion of the face. So, this parameter could enhance algorithm's strength to focus on a person who is happier than others.

5.2.1 Distance estimation

Calculating and estimating the distance between the objects (here object is a face) and camera is a main challenge for lots of computer vision subjects and - recently - in 3D computer games. Actually, lots of proposed methods for depth estimation are based on zooming and focusing. Hence, most of the cameras that have the ability of estimating the depth are using sensors and zoom lens system. However, zoom lens systems have some difficulties for modelling because they contain complex optical properties [22].

Distance estimation difficulties:

Measuring the distance between object and camera is a complicated subject which can be solved by different algorithms. Usually, some methods use more than one camera or different types of sensors to estimate the distance and some other methods based on using audio signals to calculate it. However, using advanced camera is very common these days as it has been used for FurHat before also. All of mentioned methods require some extra devices or equipment which means that the processing time might be change/increase according to use parallel device or additional calculation. In this level, we are just looking for a simple way to estimate the distance in order to find the output which can support our time limitation also. Modifying this subject can be done in future work.

Distance estimation solution

Since, the important factor for the output is being real-time in this stage and of course the main goal of this project is not focus on finding the distance, calculating the distance between the face and camera is replaced with estimating it by measuring the face's area. The bigger face's area represents the less distance and it means that the face is nearer to the camera.

For finding the area of each face, access to the border of face is mandatory. In the outputs of SHORE software, four coordinates provide the corners of

face region with an acceptable level of accuracy; top, bottom, left and right coordinate of the rectangle around the face.

The concept of detecting the depth is just comparing the distance and the target is detecting the nearest face, so, using this method will not cause a main error in the calculation.

5.2.2 Angels with the camera

In the case of FurHat, camera has a role of person who wants to talk to the other persons. So, every faces in front of the camera would have an angle with the center of the camera. If nose is assumed as a middle of the face, it is possible to calculate the angle of this point to the center of camera in "X" and "Y" directions in screen plane. "X" and "Y" equal "ZERO" means the center of face is exactly coincided with the center of the camera. These parameters could be useful for ignoring the faces that have larger angles and located in the corners. At least they have less priority to choose as an interested person rather than those who are closer and near to the middle.

5.2.3 Face angle (head rotation)

When people are talking to each other, normally, they are face to face and when they are not exactly in front of each other (for example when somebody is setting beside his/her friend), they turn their head (face) to the other person to have more eye contact. So if we want to use this pattern in human robot interaction, it is required to use the face angles. As mentioned before, head can rotate in three different directions "Roll", "Yaw" and "Pitch". Roll represents in-plane rotation of the head, pitch represents the tilt of the face and yaw can be shown by nodding. Picture below demonstrates these cases clearly.

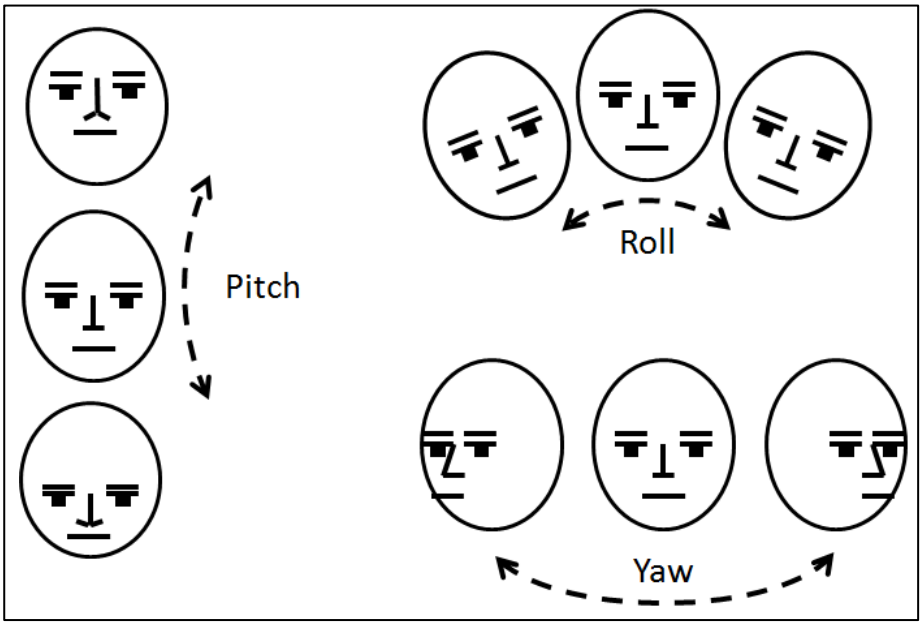


Figure 5.2. Head angels; Pith, Roll and Yaw[15].

Difficulties of calculating the face angle

Considering all three rotations at the same time in any face detecting or face tracking algorithm can definitely increase the performance time by involving all faces in different conditions. However, adding these facilities will increase the false detection also.

Solution for calculating the face angle

SHORE has three different models for detecting the face in the main algorithm. One of them is "Face.Front" that detects the face without considering the rotation, which means only faces that exist exactly in front of the camera and has none of the mentioned angels can be detected (upright frontal faces). The other one is "FaceRotated" that has the ability to recognize face with -60 to 60 in-plane rotation. In this case, the "Roll" parameter is represented a value as in the result and this value could be one of these digits: -60, -45, -30, -15, 0, 15, 30, 45, and 60. The third model is "Face.Profile" that can support detecting algorithm in different cases (upright frontal faces and out of plane rotated faces).

[15] http://msdn.microsoft.com/en-us/library/jj130970.aspx

The first one does not support our purpose so the selected item should be between second and third models. As the third one will increase the false detection and decrease the detection rate significantly, the second model has been selected until the main software will be upgraded and covered this weak point. It is obvious that the roll factor is not the parameter that we are expecting, but at least in this case, some kind of face rotation is considered. Another benefit is that, the next step - machine learning process - will be repeatable by replacing roll value with the yaw factor. Therefore, Face.Rotated model with "Roll" output is chosen in order to continue other steps.

5.2.4 Emotion

When the number of audience is more than one, electing one of them as a specific addressed can be followed with emotionally interested parameters. In other words, it is more probable that the person who is happier (and/or less angry) has chosen between two choices which are almost same in other factors. So being happy - one of the SHORE software's outputs parameters - will be noticed in next levels. By assuming this factor, the conversation could become more natural. Needless to say that it is possible to change it to the other emotional factors if it is required.

5.3 Implementations

As mentioned before, the aim of the project is to establish a method that has the possibility of starting a natural real-time interaction between FurHat and multi-person. So, the rest of this study will be focused on realizing a function which selects one specific face in each frame. Hence, some parameters are mentioned and considered to support this objective and the task's goal is selecting the right person who is more interested to start talking in each frame. The chosen parameters will guide the robot to follow this rule and, surly, it is possible to add some parameters to this function or remove those that will have no Effectiveness. However, the definition of priority function could be useful in this investigation for our purpose.

As shows in figure (5.3) a face has a square border. Corners coordinate is determined in the output. To calculate the area, it is enough to find the

length (by subtracting right and left coordinate) and the width (by subtracting bottom and top).

$$\text{Face Area} = \text{Length} * \text{Width} = (\text{right} - \text{left}) * (\text{bottom} - \text{top})$$

The area's value is located with red digits in the middle of the face and as it is expected it could help the algorithm to approximately detect which face is closer to the camera by comparing these values for all faces in the same frame.

Two blue digits in the upright side of the face border represent the angle of the face with the middle of the camera. The angle of the middle the face (the coordination of the nose is considered for the middle of the face) with the camera is calculate by considering the length and width of the camera picture and the angles

Every camera has a specific field of view which has a maximum value in both horizontal and vertical directions. Fx is the maximum angle that the camera can support in horizontal direction and Fy is the maximum angle that the camera can support in vertical direction. Also, X and Y are the numbers of pixels in horizontal and vertical direction respectively. The x and y values represent the coordination of the nose (middle the face). So, the angles could be calculated as below:

$$x1 = x - X/2;$$

$$y1 = y - Y/2;$$

$$xangle = (Fx/X) * x1;$$

$$yangle = (Fy/Y) * y1;$$

Figure 5.3. Display an output on one frame of live video.

5.3.1 Machine learning algorithm

The aim of this context is concentrated on providing a method to calculate coefficients for a function that can help and decide for the robot which face should be chosen to track in specific frames. As explained before, just some factors are considered in this level of research and working on others can happen in future investigations in this area.

By measuring the required parameter, the following equation would support the goal:

Priority = A*area + B*camera angels + C*emotion + D*head angel

Parameters A, B, C and D would be a set of coefficients which are the results of machine learning algorithm and they could guide the robot to act naturally. As the "X" and "Y" parameters of camera angle really comes together and in general has the same level of priority, only one parameter (B) has been chosen for this factor. The basics of the method that is used for the machine learning part will be described in the algorithm part. The Matlab Code is in appendix B.

Describe the Algorithm

The first step for having the "priority function" is to define an algorithm which can provide the coefficients and then test the result to realize how much they are correct. So, live movies or saved pictures are needed for preparing the data base.

Here, the data base is contains 100 images which have been taken from a movie (Friends Series). The main reason that using images prefers (instead of live movie) is that in this case the experiment is repeatable and at least the results can be checked in other steps. In live movies, it is almost impossible to train data several times by exactly same pictures. 90 pictures are chosen for the training and 10 pictures for labelling the results.

The test starts from running the C++ code in order to choose the faces in every frame according to human being's idea. This is done by moving the mouse on to the specific face. That specific face will be selected because from the user's point of view this person is more interested to start a conversation. Or even maybe the user is interested to start to talk with that specific person. In this research, there process is repeated four times by one person. However it would be better if it has been done by different persons.

By moving the mouse on the right face digit "1" has been written in front of that face information in a text file in the C++ code's results and this will repeat for every picture.

It means that by running the C++ code for every 90 images, user should select the face that is more probable to start a conversation. Certainly, the result will depend on the person who is selecting the faces. However by defining the above parameters and describing the reason that they are used we try to make it clearer and more similar.

The next stage is running the Matlab code. The code is containing the comparison method and machine learning part in order to detect the face by

the machine. It means that the priority function, which describe above, will provide the output for each frame in every set of images (90 images) by generating random coefficients between -1 and 1 for A, B, C and D. The coefficient sets should have compatibility with the highest percentages of correct selection.

It means that one part of the Matlab code is focused on generating different sets of random coefficient for the priority function. So, in every loop, code is going to select a random set for A, B, C and D. By these four factors and also having the area, camera angles, emotion and head angle, we can have a value for every face in every picture as a priority value.

Priority = A*area + B*camera angels + C*emotion + D*head angel

The highest priority value in each frame will represent the selected face from machine point of view and digit "1" will be written in front of the selected face.

Another part of the code is comparing the results of those two methods; selected face by user and selected face by the machine in every frame. If they are same, the algorithm will return "1" and in case they are different faces it will return "0".

The output becomes a number between 1 to 90 which will demonstrate in how many frames the selected face (by the machine) is exactly the same with that one selected by the user. For instance, digit 59 in the output means that in 59 images the face that user selects is the same face that has highest value according to the Priority function. As the number of image set is 90, the percentage becomes 65.5%.

(59*100)/90 = 65.5

This whole procedure should run one more time for rest 10 test images to check the results. In this point, again the percentage of the correct selection for every set of parameters is the main factor should be considered.

This processes has repeated 100, 000 time. In the next chapter, more details are presented about the results and algorithm. Needless to say, the aim of this research is focus on the method not on the output's digits as it can be modified by increasing the number of parameters and using advanced machine learning methods.

6 Results

Previously, the main process is explained in different stages. So, the results could include the whole process by repeating 100, 000 times with random parameters and it will be presented in this chapter.

After running the Matlab code, we would have a text file with 100,000 lines and 5 columns. From left to right these columns are A, B, C and D coefficients of the priority function and the fifth column shows the number of frames that the selected face by user and code are similar (the first 5 columns of table 6.2. is an output sample). For example, the number in the fifth column equals to 63 means that in 63 images (out of 90 images) the chosen face by the algorithm is exactly the same as the chosen face by the human and it represents the value of the priority function for that specific face is bigger than other faces in that frame.

In the final results, the maximum match case is 79 and the minimum one is 36. In the table 6.1. the frequency distribution of data is demonstrated for the number of similarity more than 70. According to the table, 67 times (out of 100, 000 times) the number of similarity becomes maximum which means that in 79 pictures the human's and machine's selected faces are the same. While the input parameters are selected randomly, these frequency distribution digits indicate any specific.

Since the aim of algorithm is finding the highest similarity between the proposed methods, we are going to ignore some parts of the bad data and consider those parameters which can prepare better results. So, those sets of parameters that have 78 similar cases or more are selected after sorting the fifth column. More than 78 similar cases out of 90 means the percentage should become about 86.6%. The only reason that 78 similarity (or 86.6%) is chosen for the final test is that the number of match cases equals to 77; as it can be seen it is 12486, and since we are looking for some sets of coefficient which guide us to the some decisions, this large number of coefficient sets will avoid the results to converge. Therefore, only last two top similarities are selected (number of similarity equals 78 and 79) which means that for testing the 10 remaining images in the data base the algorithm repeated 943 (= 876+67) times.

Number of similarity	Frequency Distribution (in 100,000 times)
70	127
71	105
72	163
73	162
74	421
75	2682
76	32320
77	12486
78	876
79	67

Table 6.1. Frequency Distribution for similarity more than 70.

By running the same code for remaining 10 images in the data base with these 934 selected coefficients, it becomes obvious that which sets will come up with the better result. The comparison shows that the minimum match case is 7 while the maximum one is 9. Actually, more than 70% accuracy demonstrates the chosen sets are almost good. However, (by checking all sets in table 6.2. we find out) none of those sets which equals to 9 is in the group of 79 match cases in the first round and all of them is in 78 mach cases. In the table 6.2. these coefficient sets are presented.

As all this outputs have the same conditions and the source generator is random, it will be obvious that there is no priority between them. As a matter of fact, the aim of this text is not finding a set of coefficient for priority function. The main aim is to provide and suggest a method to find a priority function for human robot interaction and, of course, this method should be improved in the future researches.

Coeff. A	Coeff. B	Coeff. C	Coeff. D	No. of similarity out of 10 images	No. of similarity out of 90 images
0.082	0.715	-0.604	-0.689	9	78
0.068	0.543	-0.524	0.251	9	78
0.118	0.813	-0.885	0.359	9	78
0.107	0.758	-0.871	0.783	9	78
0.059	0.465	-0.468	-0.444	9	78
0.081	0.682	-0.628	0.970	9	78
0.115	0.999	-0.963	-0.730	9	78
0.070	0.584	-0.535	0.476	9	78
0.099	0.920	-0.705	-0.192	9	78
0.115	0.908	-0.937	-0.162	9	78
0.085	0.696	-0.622	-0.246	9	78
0.058	0.517	-0.493	0.800	9	78
0.020	0.286	0.070	0.636	9	78
0.108	0.908	-0.825	0.833	9	78
0.076	0.696	-0.604	0.454	9	78
0.105	0.876	-0.850	-0.519	9	78
0.074	0.891	-0.551	-0.652	9	78
0.051	0.399	-0.401	-0.427	9	78
0.038	0.371	-0.263	-0.199	9	78
0.064	0.713	-0.487	0.045	9	78
0.016	0.144	-0.125	-0.313	9	78
0.065	0.719	-0.504	0.480	9	78
0.046	0.286	-0.348	0.115	9	78
0.104	0.764	-0.827	-0.315	9	78
0.047	0.390	-0.366	0.279	9	78
0.034	0.225	-0.276	0.624	9	78
0.076	0.823	-0.573	0.032	9	78
0.065	0.736	-0.534	0.805	9	78
0.086	0.895	-0.622	0.548	9	78
0.064	0.604	-0.504	0.912	9	78
0.103	0.968	-0.788	-0.592	9	78

Table 6.2.Sets of coefficient which provides the highest similarity.

7 Discussion and suggestion

7.1 Discussion

Although SHORE algorithm is the best real-time multi-person tracker among all the studied, it is still possible to attempt more in order to find a better one. According to some errors that occur in detecting the face in this algorithm, it might be beneficial if it can be replaced by the modified version or another algorithm with less error. However, the errors could have negligible effects on the results - priority function - by choosing the suitable sample pictures for data base; for example with less complexity in the background. In reality, the background for the usage of FurHat is might be a wall and it will not very messy.

Estimating the depth in image processing is complex and sometimes might take time in the elementary algorithm. Using more advanced methods to measure the distance between the person and camera may increase the running time of the algorithm; however it could improve the results to the more correct ones.

Chosen parameters for calculating the priority function is based on the available output parameters from SHORE software. These selected parameters may not cover the whole criteria that we are looking for, but they are almost the best ones in this stage of the project.

The machine learning method that is used for finding the priority function can be replaced with an advance one. According to the time and expected results for this stage of project, one of the simplest methods has been chosen for the machine learning algorithm. Certainly, the output will be more accurate by using more advanced methods.

By using this method, it is almost impossible to choose one set of outputs as a final coefficient. In order to have a specific and useful result, more statistical researches are required.

As mentioned before, considering the yaw angle - among all three different rotated angels in the face (pitch, roll and yaw) - would be more effective in the calculation, but because of some limitations in the version of the software we have to use a replacement. This replacement does not have any effect on the structure of priority function or machine learning algorithm. However, changing “Roll” with “Pitch” could guide us to better result.

7.2 Suggestion

As a result of first part of this investigation, SHORE software was found out. Although this method can change FurHat's future, searching for other methods is still a way of prepare a good interaction for this robot.

Due to lack of time, the machine learning methods that is used in this project is based on a simple method. To enhance the accurate level of the outputs, using other methods and compare the results is strongly recommended.

The final results are completely raw and their needs more statistical analysis to compare different sets of outputs and choose the best one.

Among all the parameters which are considered in priority function, estimating the depth and recognizing the yaw angle would have more effect to detecting the right face. By having the most accurate values for these two parameters, the error will increase significantly.

As regards all those parameters which has been used in the priority function, there is most likely to improve the result by considering some new factors or removing the current ones. Surely, there are no criteria for detecting the best set of parameters and it will happen by trial and error processing.

8 Conclusion

The goal of this project is to figure out a way which can provide a real-time interaction between humans and robots. Moreover, the method should have minimal limitation in the number of detected face and/or the distance between the robot and human. This algorithm will be used by the head robot which is designed in KTH - FurHat- in order to support real-time multi-person conversation.

FurHat currently uses advanced camera (Kinect camera) in order to take part in a conversation. According to the limitations of this device the research group suggest some methods to replace with this one. Colour tracking and integrating Kinect camera with some other solution was two other options at the beginning, but the most focus was on a method which uses face detection algorithms.

In face detection area the first idea comes from the market offers, SHORE software, which covers most of the requirements of this interaction. This is one of the productions of Fraunhofer Company and they accepted to provide the source of the software for academic aspects.

Next step is defining an interaction way between software and robot. The result is based a function which is supposed to select one face in every frame by considering who is more interested to start a conversation. This function has found out by integrating the available parameters from the software and the analysis data with a machine learning algorithm.

The parameters which are used for determining the function is:

1) Area: representing the distance between the camera and person (depth)
2) Face angel with the camera: two angles in X and Y direction in the camera plane to ignore that ones who are in the corners
3) Head rotation: the angle of face with the neck (pitch or yaw or roll)
4) Emotion: which has the value if the person has smile on the face

Certainly, the result is not centralized on finding one specific answer for the priority function. The main idea is just providing a method to settle the problem.

8.1 Future works

The face detection topic is covering a wide range of researches and it is almost impossible to collect all related knowledge in one specific category. Here is some suggestion for future works for this investigation. Of course future works in not limited to these items.

The distance between the person and robot is estimated by the value of face area in this project. This is just valid until we accept errors in our results depend on accuracy level and/or its function. For having accurate output detection it is required to focus more on the details. However, precise depth detection will increase run time of the algorithm.

Replacing the head rotation factor “yaw” instead of “Roll” helps the algorithm to consider a more correct head rotation in calculation and cause better results. In this case, the probability of selecting a person who is in front the camera but not looking at it becomes low.

Coefficients for the function are the output of the MATLAB program which is selecting them through a random function and repeat the algorithm 100,000 times. This method is providing different sets of coefficients which can be use by the robots. However, choosing a more enhanced method for finding coefficients would decrease the error and make the final function more strong.

The data base has a main role in the results. In this project the goal is selecting a very random data base in order to settle the main method. It is obvious that using another data base would change the results. The suggestion is testing different types of data base and compares the results to obtain the best one.

Integration the result of this project by audio signal which is provided by the person who wants to talk to the robots can prepare a robust method to have interaction with less error.

9 References

[1] Al Moubayed, S., Beskow, J., Skantze, G., & Granström, B. (2012). Furhat: a back-projected human-like robot head for multiparty human-machine interaction. In Cognitive Behavioural Systems (pp. 114-130). Springer Berlin Heidelberg..

[2] Jun, B., & Kim, D. (2007). Robust real-time face detection using face certainty map. In Advances in Biometrics (pp. 29-38). Springer Berlin Heidelberg.

[3] Verschae, R., Ruiz-del-Solar, J., & Correa, M. (2006). Gender classification of faces using adaboost. In Progress in Pattern Recognition, Image Analysis and Applications (pp. 68-78). Springer Berlin Heidelberg.

[4] Zhang, C., & Zhang, Z. (2010). A survey of recent advances in face detection. Tech. rep., Microsoft Research.

[5] Caleanu, C. D., & Botoca, C. (2007). C# solutions for a face detection and recognition system. Facta universitatis-series: Electronics and Energetics,20(1), 93-105.

[6] Wu, J., Brubaker, S. C., Mullin, M. D., & Rehg, J. M. (2008). Fast asymmetric learning for cascade face detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(3), 369-382.

[7] Yang, M. H., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on,24(1), 34-58.

[8] Kim, H. D., Choi, J. S., & Kim, M. (2007). Human-robot interaction in real environments by audio-visual integration. INTERNATIONAL JOURNAL OF CONTROL AUTOMATION AND SYSTEMS, 5(1), 61.

[9] Tsai, C. Y., Dutoit, X., Song, K. T., Van Brussel, H., & Nuttin, M. (2010). Robust face tracking control of a mobile robot using self-tuning Kalman filter and echo state network. Asian Journal of Control, 12(4), 488-509.

[10] CL, S. N., NIKHRA, V., JHA, S. R., DAS, P. K., & NAIR, S. B. AN INTELLIGENT FACE TRACKING SYSTEM FOR HUMAN-ROBOT INTERACTION USING CAMSHIFT TRACKING ALGORITHM. International Journal, 3.

[11] Pateraki, M., Baltzakis, H., Kondaxakis, P., & Trahanias, P. (2009, May). Tracking of facial features to support human-robot interaction. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on (pp. 3755-3760). IEEE.
[12] Littlewort, G., Bartlett, M. S., Fasel, I. R., Chenu, J., Kanda, T., Ishiguro, H., & Movellan, J. R. (2003). Towards Social Robots: Automatic Evaluation of Human-robot Interaction by Face Detection and Expression Classification. In NIPS.
[13] Luxand FaceSDK Documentation, "Luxand FaceSDK 4.0 Face Detection and Recognition Library," Developer's Guide, Copyright © 2005–2011 Luxand.
[14] myAudience-Measure Product Overview, "myAudience-Measure Complete solution for automated audience measurement research," Copyright © 2011 Rhonda Software.
[15] VeriLook SDK Brochure 2012, "VeriLook SDK Face identification for PC or Web applications," Last Updated 2012.
[16] VeriLook Surveillance SDK Brochure 2012, "VeriLook Surveillance SDK Face identification for video surveillance systems," Last Updated 2012.
[17] MegaMatcher_SDK_Brochure_2012, "MegaMatcher SDK Large-scale AFIS and multi-biometric identification," Last Updated 2012.
[18] Bolme, D. S., Beveridge, J. R., & Draper, B. A. (2009). Facel: Facile face labeling. In Computer Vision Systems (pp. 21-32). Springer Berlin Heidelberg.
[19] faceAPI Brochure, "faceAPI The Real-Time Face Tracking Toolkit for Developers and OEMs," © Copyright 2008.
[20] Küblbeck, C., & Ernst, A. (2006). Face detection and tracking in video sequences using the modifiedcensus transformation. Image and Vision Computing, 24(6), 564-572.
[21] SHORE user manual, Version 1.4.0 Generated by Doxygen 1.6.3, 2011.
[22] Baba, M., Asada, N., Oda, A., & Migita, T. (2002). A thin lens based camera model for depth estimation from defocus and translation by zooming. InProceedings of (Vol. 107, pp. 274-281).

Appendix A

Definition of Parameters for Fraunhofer Software (SHORE)

timeBase Video sampling interval in [s]. 0 means video filtering off (single image mode). E.g. a video of 25 frames/second would require a value of 0.04 (40ms). If updateTimeBase is true then the value given here (if different from 0) is only used as a first estimate and is updated automatically during runtime. The valid range of this parameter is [0, 10].

updateTimeBase If updateTimeBase == true the automatic estimation and adaption of the current time base is turned on. Otherwise turned off. This parameter takes only effect if timeBase > 0 (video mode). See also the timeBase parameter.

threadCount The number of threads used for processing. If you run your application on a machine with more than one single cpu, it typically will be faster to use all cpus. Notice that the result of processing an image can be slightly different from the single threaded version. The number of threads is limited to 10. But it only makes sense to provide at most the number of available cpus available.

modelType Defines which model is used to detect faces. Valid values are ’Face.Front’, ’Face.Rotated’ and ’Face.Profile’. The returned face objects also have the three attributes with key ’Roll’, ’Pitch’ and ’Yaw’. The ’Roll’ attribute describes roughly the in-plane rotation of the face. The ’Pitch’ attribute indicates the tilt of the face. The pitch of a face can be compared with nodding. The ’Yaw’ attribute describes roughly the out-of-plane rotation, e.g. for profile faces the yaw is ’-90’ and ’90’ respectively.

For the value ’Face.Front’ a model is used to detect upright frontal faces. This requires the model ’FaceFront_24x24_2008_08_29_161712_7.cpp’ to be added to the project and registered. All returned face objects will have the value ’0’ for the ’Roll’, ’Pitch’ and ’Yaw’ attributes. The face objects also have two markers with key ’LeftEye’ and ’RightEye’ that describe roughly the position of both eyes.

Providing ’Face.Rotated’ as value, a detection model is used that is able to detect -60 to +60 degree in-plane rotated faces. This requires the model ’FaceRotated_24x24_2008_10_15_-180432_24.cpp’ to be added to the project and registered. For rotated face objects the attribute ’Roll’ contains the in-plane rotation (’-60’,’-45’, ’-30’, ’-15’, ’0’, ’15’, ’30’, ’45’, ’60’). The ’Pitch’ and ’Yaw’ attributes will have the value ’0’. The rotated face objects also have two markers with key ’LeftEye’ and ’RightEye’ that describe roughly the position of both eyes.

For the value ’Face.Profile’ a model is used that is able to detect upright frontal and out-of-plane rotated faces. This requires the model ’Face_24x24_2009_09_02_185611_48.cpp’ to be added to the project and registered. The ’Yaw’ attribute contains the out-of-plane rotation (’-90’, ’-45’, ’0’, ’45’, ’90’). In addition the model also detects profile faces (yaw equal to ’-90’ or ’90’) that are pitched. Possible values for the attribute ’Pitch’ are ’-20’, ’0’ and ’20’. For the ’Yaw’ values ’-45’, ’0’ and ’45’ the face objects have the three markers with key ’LeftEye’, ’RightEye’ and ’NoseTip’ that describe roughly the position of both eyes and the nose tip. If the value of the ’Yaw’ attribute equals ’90’, the face objects have three markers with key ’LeftEye’, ’NoseTip’ and ’LeftMouthCorner’. For the value ’-90’ of the ’Yaw’ attribute the face objects have three markers with key ’RightEye’, ’NoseTip’ and ’RightMouthCorner’.

Keep in mind that the rotated and profile models do not have the same performance than the upright frontal model (lower detection rate and more false detections). And please consider that the analysis and fine search modules only act on face if the ’Roll’ attribute equals ’-15’, ’0’ or ’15’, the ’Pitch’ attribute equals ’0’ and the ’Yaw’ attribute equals ’-45’, ’0’ or ’45’.

imageScale This parameter resizes the original input image using this scaling factor for the internal processing. The valid range is in [0,3]. The value 1 leaves the image in the original size. This parameter has two purposes. On the one hand you can use it to downscale the input image. This can make sense if you are only interested in faces bigger then 24x24 pixel and want to speed Generated on Tue Jan 11 14:35:40 2011 for Shore by Doxygen18 Namespace Documentation up the detection. On the other hand this parameter can be used to upscale the input image to

Find faces even smaller then 24x24 pixel. Pay attention that downscaling the input image can have drawbacks on the detection performance. If you want to find only bigger faces with the full detection performance it is

recommended to limit the minimal face size via the minFaceSize parameter. If speed is important downscaling the input image could be a good choice. To detect faces smaller then 24x24 pixel the only option is to upscale the input image. Please pay attention that the minFaceSize parameter must be set to an appropriate value in this case. For example if

you want to find faces with at least 12x12 pixel in size, you must set the image scale to 2 and the minimal face size at most to 12*12, 0 or a proper relative value depending on the input image size. The input image must be sharp for good results on small faces. See also the minFaceSize parameter documentation.

minFaceSize This parameter limits the minimal face size that will be returned. It must be greater or equal to 0. If it is in the range [0, 1] it is taken with respect to the original image size. If it is greater than 1.0 it is the absolute minimal number of pixels a face will have in the original image. Thus if you want the minimal face size to take at most a quarter of the image area you have to provide 0.25 for this parameter. If you want to detect faces with a minimal size of about 100*100 pixels in the image just provide 100*100 for this parameter. Regard that this parameter is only the lower bound for the face size. The faces actually returned by the engine will normally be slightly bigger. So choose this parameter generous. Although you can provide 0 as the face size, the minimal detected face size is limited by the used model size which is at the moment 24x24 pixel. But you can set the imageScale parameter to 3 to find even faces of 8x8 pixel in size. Consider that limiting the minimal face size often speeds up the detection process without drawbacks on the detection performance. See also the imageScale parameter documentation.

minFaceScore This is the minimal face detection threshold. Each face has a rating with key ’Score’ that indicates how likely it is to be a face. Using this parameter you can set a minimal score for the faces that will be returned by the engine. The lower this value is the more faces but also false detections will be found. A good choice for the model type ’Face.Front’ and ’Face.Rotated’ is 9 and for ’Face.Profile’ 16. A reasonable range for all models is in [0, 60].

idMemoryLength This parameter takes only effect if timeBase > 0 (video mode). As already men-tioned above, in the video mode an identity number is assigned to each face in the object attribute with key ’Id’ which is constant as long as the trajectory can be determined without discontinuity. This parameter defines how long (in seconds) a face is remembered internally, so that a connection of a currently detected face to the same face detected some time ago can be made. How this connection is made and which information is returned by the engine is defined by the parameter idMemoryType. The valid range of idMemoryLength is [0, 180].

idMemoryType This parameter takes only effect if timeBase > 0 (video mode) and if idMemoryLength > 0. In this case valid arguments are "Spatial", "Recent" or "All". Their meaning is described in the following. For "Spatial" the face id is remembered internally and assigned again if a face is lost and found again at approximately the same position in the image. However if a face is lost and found again in a different position a new id is assigned. This is the fastest method of the three.

For "Recent" and "All" ’FaceFrontId_36x44_2009_08_07_122105.cpp’ must be added to the project and registered. In both cases ’fingerprints’ of the faces are temporarily remembered and used to create links between newly detected faces and faces already detected some time ago. Therefore it can be quite a time and memory consuming task, if there appear a lot of new faces in the image. The more and longer the fingerprints of the faces are stored, the more time consuming this task can be. Thus keep idMemoryLength as low as reasonable. The number of faces currently remembered is available in the content info with the key ’GallerySize’. The number of views of currently memorized faces in the gallery is available in the object attribute with the key ’ViewCount’. If idMemoryType is "Recent" and links to previously detected faces has been found the object gets attributes with key ’RecentId_0’ to ’RecentId_N’ that contain the id or ids of the faces when they were detected the last time. In case of "All" the object gets an attribute with key ’PreviousIds’ that contains a space separated list of not only the last ids Generated on Tue Jan 11 14:35:40 2011 for Shore by Doxygen5.1 Shore Namespace Reference 19 (as in case of ’Recent’ but ALL the ids this face had when it was previously detected. Take care that this is dangerous in a static environment where some insistent false detections appear again and again! In this case the list may become very very long. (To avoid this see parameter phantomTrap) Prefer

the ’Recent’ setup if possible, because it remembers and returns only the ids when the face was detected the last time. In both cases it may take some time or frames until links can be established.

trackFaces This parameter takes only effect if timeBase > 0 (video mode). If it is true an object tracker is activated that tries to track the faces that were lost by the face detection module. In this case the object type is changed to ’Face.Tracked’. Take care that the fine search and analysis modules below do not act on tracked faces. If this parameter is true ’Tracking_36x36_2008_-08_21_110315.cpp’ must be added to the project and registered.

phantomTrap This parameter takes only effect if timeBase > 0 (video mode). The module tries to find false detections (phantoms) which are left over by the previous modules. It is currently designed for static mounted cameras. The default mode is ’Off’. Usage of this module requires the model ’Tracking_36x36_2008_08_21_110315.cpp’. It have to be added to the project and registered.

Following modes are possible: ’Off’, ’Delete’, ’Mark’

- ’Off’ : the module is not used
- ’Delete’: false detections are identified and deleted on the fly.
- ’Mark’: identified false detections are marked by the module. Each object will have an additional attribute with the key ’Phantom’. The value can be ’Yes’ or ’No’. No objects are deleted in this mode. This option makes only sense, if you postprocess the added metadata.

searchEyes If true an eye fine search module is added to the engine. This requires the models ’LeftEyeFront_16x16_2008_10_20_190938_4.cpp’ and ’RightEyeFront_16x16_2008_10_20_190953_4.cpp’ to be registered. The eye fine search works better if the face has a minimal size of about 36x36 pixel in the image (for the used models). If the eye fine search was successful, the appropriate face object will get a part named ’LeftEye’ and ’RightEye’ respectively. The marker positions ’LeftEye’ and ’RightEye’ of the parts will also be transfered to the face object. Note that if you use one of the additional analyzer modules you should also turn the eye fine search on to get better results.

searchNose If true a nose fine search module is added to the engine. This requires the model 'NoseFront_16x16_2008_10_17_134731_4.cpp' to be registered. The nose fine search works better if the face has a minimal size of about 36x36 pixels in the image (for the used model). If the nose fine search was successful, the appropriate face object will get a part named 'Nose'. The marker 'NoseTip' of the part will also be transfered to the face object.

searchMouth If true a mouth fine search module is added to the engine. This requires the model 'MouthFront_16x14_2008_10_20_190419_4.cpp' to be registered. The mouth fine search works better if the face has a minimal size of about 36x36 pixels in the image (for the used model). If the mouth fine search was successful, the appropriate face object will get a part named 'Mouth'. The markers 'LeftMouthCorner' and 'RightMouthCorner' of the part will also be transfered to the face object.

analyzeEyes If true a analysis step of the eyes is added to the engine. This requires the models 'LeftEyeClosed_16x16_2008_10_23_185544.cpp' and 'RightEyeClosed_16x16_2008_10_23_185544.cpp' to be registered. Each face object will get two ratings with the keys 'LeftEyeClosed' and 'RightEyeClosed' which indicate how closed the eyes are. The ratings are both in the range [0, 100]. The higher the ratings, the more the eyes were classified as closed. The rating for open eyes is 0. It is recommended to turn the eye search on.

analyzeMouth If true a mouth analysis step is added to the engine. This requires the model 'MouthOpen_16x14_2008_10_23_185229.cpp' to be registered. Each face object will get a rating with key 'MouthOpen' that indicates how wide the mouth is open. The rating is in the range [0, 100]. The higher the rating, the more the mouth is open. The rating for a closed mouth is 0. It is recommended to turn the eye search on (mouth search is not necessary in this case). Generated on Tue Jan 11 14:35:40 2011 for Shore by Doxygen20 Namespace Documentation

analyzeGender If true a gender analysis step is added to the engine. This requires themodel 'Gender_26x26_2008_09_04_174103.cpp' to be

registered. Each face object will get an attribute with key ’Gender’ that can be ’Female’ or ’Male’ (and also “empty” in the video mode depending on the result). In the video mode this attribute is filtered over time. Furthermore the face gets two ratings with the keys ’Female’ and ’Male’, which indicate whether it is more likely to be a female or male face. The ratings are both in the range [0, 100]. The higher the ratings, the more it was classified as male or female. Pay attention that the gender model was trained mainly for European adult faces at the moment. It is recommended to turn the eye search on.

analyzeAge If true a age estimation step is added to the engine. This requires the model ’Age_28x28_2009_09_17_131241.cpp’ to be registered. Each face object will get two additional ratings with the keys ’Age’ and ’AgeDeviation’. The value of the ’Age’ rating is the estimated age in years, which ranges in most cases from 0 to 90 years. The ’AgeDeviation’ rating indicates the deviation (mean absolute error) of the age estimation on two public available test sets. For more information about the test sets please contact us. Considering the estimated age and the deviation a believable range for the real age can be defined by estimated age +/- deviation. Pay attention that the age estimation was trained mainly for European faces at the moment. It is recommended to turn the eye search on.

analyzeHappy If true a happy analysis step is added to the engine. This requires the model ’Happy_26x26_2008_09_08_124526.cpp’ to be registered. Each face object will get a rating with key ’Happy’ that indicates how much it is the case. The rating is in the range [0, 100]. The higher the rating, the more it was classified as happy. It is recommended to turn the eye search on.

analyzeSad If true a sad analysis step is added to the engine. This requires the model ’Sad_26x26_2008_10_21_161703.cpp’ to be registered. Each face object will get a rating with key ’Sad’ that indicates how much it is the case. The rating is in the range [0, 100]. The higher the rating, the more it was classified as sad. It is recommended to turn the eye search on.

analyzeSurprised If true a surprised analysis step is added to the engine. This requires the model 'Surprised_26x26_2008_09_11_175815.cpp' to be registered. Each face object will get a rating with key 'Surprised' that indicates how much it is the case. The rating is in the range [0, 100]. The higher the rating, the more it was classified as surprised. It is recommended to turn the eye search on.

analyzeAngry If true a angry analysis step is added to the engine. This requires the model 'Angry_26x26_2008_10_21_152601.cpp' to be registered. Each face object will get a rating with key 'Angry' that indicates how much it is the case. The rating is in the range [0, 100]. The higher the rating, the more it was classified as angry. It is recommended to turn the eye search on.

Appendix B

```
clc;clear;
tic;
fid = fopen('D:\CmdLine\example.txt','r+');

% read column headers
C_text = textscan(fid, '%s', 10,  'delimiter', '\t');

% read numeric data
C_data0 = textscan(fid, '%f %d %d %d %d %f %d %d %d %d %d');
fclose(fid);
output=zeros(size(C_data0{1,1},1),1);

fid1 = fopen('d:\zahra\facedetect -
shore\demo\cmdline\example2.txt','a+');
for factor=1:100000

    % Creat random data
    x1=[random('unif',-1,1), random('unif',-
1,1),random('unif',-1,1),random('unif',-1,1)];

    % Build Priority Function
    for i = 1 : size(C_data0{1,1},1)

output(i)=x1(1,1)*C_data0{1,1}(i,1)+x1(1,2)*C_data0{1,2}(i,1
)+x1(1,2)*C_data0{1,3}(i,1)+x1(1,3)*C_data0{1,6}(i,1)++x1(1,
4)*C_data0{1,7}(i,1);
    end
    C_data0{1,8}(:,1)=output;
    counter =0;


    % count the match objects and print in the output file
    for j = 1 : size(C_data0{1,1},1)
        a=j;
        while (j==1 && a<2 )
            for l=1:size(x1,2)
                fprintf(fid,'%f\t',x1(1,l));
            end
            a=a+1;
        end

        find_frame_number=find
(C_data0{1,5}(:,1)==C_data0{1,5}(j,1));
```

```
        for k = 1:size (find_frame_number)
            A(k)=C_data0{1,8}(find_frame_number(k),1);
        end
        max_priority = max (A);
        if (C_data0{1,8}(j,1)==max_priority &&
C_data0{1,10}(j,1)==1)
            counter = counter + 1;
        end
        A=0;
    end
    fprintf(fid,'%f\n',counter);

end
fclose(fid1);
fclose all;toc;
```