CHALMERS

Robust Face Recognition on Adverse 3D Data

Attaining Expression & Occlusion Invariance Using Machine Learning

Master's thesis in Complex Adaptive Systems

MIKAEL KÅGEBÄCK

Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2013
Master's thesis 2013:41

MASTER'S THESIS IN COMPLEX ADAPTIVE SYSTEMS

Robust Face Recognition on Adverse 3D Data

Attaining Expression & Occlusion Invariance Using Machine Learning

MIKAEL KÅGEBÄCK

Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2013

Robust Face Recognition on Adverse 3D Data
Attaining Expression & Occlusion Invariance Using Machine Learning
MIKAEL KÅGEBÄCK

Master's thesis 2013:41
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Average face model constructed using 3D scans of 105 individuals.

Chalmers Reproservice
Gothenburg, Sweden 2013

Robust Face Recognition on Adverse 3D Data
Attaining Expression & Occlusion Invariance Using Machine Learning
Master's thesis in Complex Adaptive Systems
MIKAEL KÅGEBÄCK
Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

The emerging field of high resolution mobile and inexpensive depth cameras, promise to revolutionize many parts of computer vision. One area in particular where 3D data has been shown to improve performance, is face recognition. Using a combination of local and global pattern matching and a committee of neural networks, this thesis present a robust 3D face recognition approach, decisively outperforming current methods. The system is evaluated on the Bosphorus database, a challenging benchmarking dataset that include face scans with both facial expressions and partial occlusions, captured in angles of up to 90° rotation. The proposed system achieves a recognition rate of 98.9%, which is the highest recognition rate ever reported on the Bosphorus database, improving the state of the art by 5.2%.

Keywords: 3D, Face Recognition, Machine Learning, Neural Networks, ICP, Computer Vision, Deep Learning

Acknowledgements

I would like thank my supervisor associate professor Krister Wolff for much appreciated help and support, and the Bogazici University for suppling me with their 3D face database. I would also like to take the opportunity to express my love and gratitude towards my family, for supporting me and showing great patience during the writing of this thesis.

Contents

Chapter 1

Introduction

The problem of biometric human identification can be approached in several different ways. Analysis of fingerprints has historically been very successful and has the advantage of being highly accurate, but it suffers the disadvantage of being quite intrusive. A non intrusive alternative to fingerprints, which has been shown to hold the same discriminative power, is face recognition [23]. Looking to nature, it turns out that the ability to perform face recognition is so important for humans, that the human brain has evolved a center, the fusiform face area, specialized for this task [17]. This tells us two things, (1) nature has deemed this ability advantageous enough to spend significant amounts of time and energy on acquiring it, and (2) face recognition is a problem that probably needs specialized processing to be solved satisfactorily.

Much research has been spent on face recognition using 2D color pictures. The motivation behind this is first, that humans are able to identify other humans from 2D pictures and hence it should be possible to program a computer to do the same. Second, until recently hardware for 3D surface acquisition has been either low resolution or expensive and bulky. However, the latter is no longer valid, which has spurred renewed interest in the use of 3D data for face recognition. Using 3D data, stronger classifiers can be constructed that are more resilient to circumstances such as lightning conditions and cosmetics, which has proved a big challenge for pure 2D techniques.

The use of 3D data in face recognition does, however, come with its own challenges. Two of the largest challenges being, (1) facial expressions that distort the global shape of the face and makes global face matching inaccurate, and (2) occluded or non frontal face scans that limit the usable facial surface and may confuse the registration[1] process.

In this thesis a novel 3D face recognition approach is presented, insensitive to both facial expressions and partial occlusions. The method is evaluated on a dataset exhibiting a full range of facial expressions, viewing angles, and partial occlusions. In this demanding setting, the proposed method outperforms all other methods found in current research.

[1] Registration is the process of transforming raw input data to a common coordinate system, see Section 1.3.3 for details.

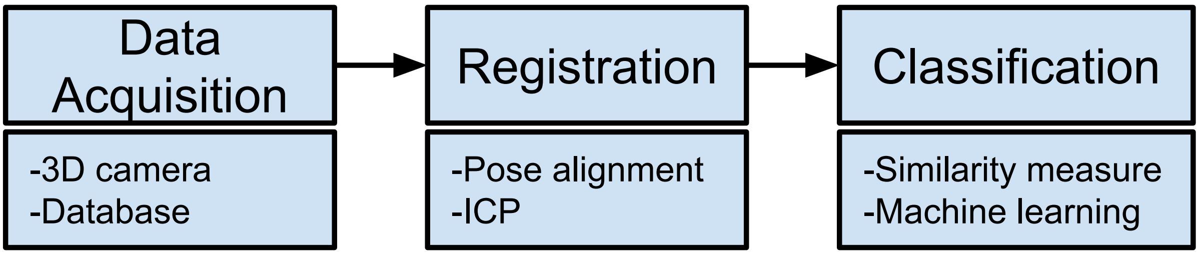


Figure 1.3.1: *Schematic overview of a generic face recognition pipeline.*

1.1 Purpose

The aim of this thesis is to investigate whether a robust face recognition system may be created, utilizing 3D surface data. To be considered successful the system should be resilient to:

1. Pose variations of up to 90° rotation.

2. Partially occluded faces in realistic situations, e.g. a hand or hair covering parts of the face.

3. A full range of facial expressions, ranging from subtle smiles to very distorted faces.

1.2 Limitations

The system will be evaluated using a benchmarking dataset of manually landmarked[2] faces, see Section 1.3.2 for details. However, the system should not depend on the exact nature of manual landmarking, but generalize to automatic landmarking.

1.3 Background and related work

A face recognition system may be decomposed into three subsystems, see Figure 1.3.1. (1) A data acquisition system that captures the facial surface, (2) a registration process that frames the raw input data in a well defined coordinate system and, (3) a classification system.

1.3.1 Data acquisition

The first step in all signal processing applications, is to digitize the physical phenomena that is to be analyzed. In the particular case of 3D face recognition, this constitutes capturing a three dimensional representation of the facial surface to be recognized. There are several techniques that can accomplish this, the three dominant ones are passive stereo vision, structured light and time of flight.

[2] A facial landmark refers to the position of a facial feature, e.g. the nose tip, the corners of the eyes, or any other distinguishable position on a face.

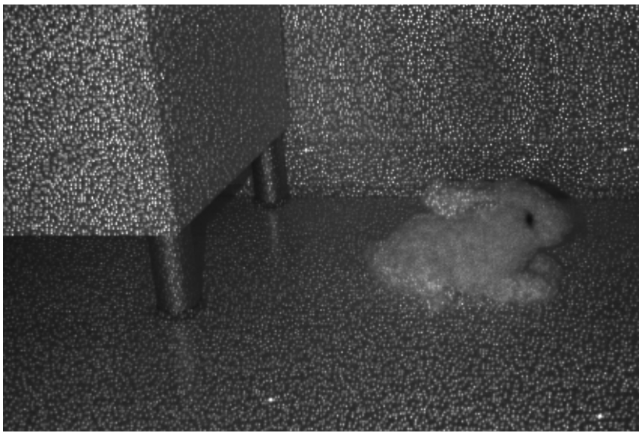


Figure 1.3.2: *IR image of the projected light pattern produced by a Kinect Sensor.*

Passive stereo vision

Passive stereo vision is a biologically inspired 3D reconstruction technique that uses two RGB cameras, mounted side by side, each capturing the scene with a slight offset to the other camera. By comparing the location of key-points found in both images it is possible to triangulate the depth of these using the known horizontal offset between the cameras.

The fact that no special hardware is needed, and that it is similar to human 3D vision, has made it popular in the humanoid robotics community. However, it has a couple of drawbacks, (1) it is computationally intensive, and (2) it fails when there is no, or too little, texture to extract common key-points from [30], e.g. very plain surfaces or bad lighting conditions.

Structured light

Structured light sensors [8] project a complex light pattern into the environment using an infrared (IR) laser projector, see Figure 1.3.2. The light pattern is subsequently captured using an IR detector, and the positions corresponding to local patterns found in both the IR image and the internal representation of the light pattern (in the memory of the sensor), is extracted. Using these positions, and the known projector to camera offset, three dimensional points on the surface of the object being captured, is triangulated.

The active nature of this sensor makes it less sensitive to structure and lighting of the object being captured. However, the fact that it is asymmetric gives rise to shadows on one side of the object where no depth information can be extracted, see Figure 1.3.3.

Time of flight

Time of Flight (ToF) cameras [8] operate using modulated infrared light, measuring the flight time by comparing the phase of the sent and reflected signal.

ToF sensors are the most expensive type of sensor investigated and produce rather low resolution images. However, they do not suffer from the problems associated with featureless surfaces inherent to passive stereo vision, or the blind spot shadows of structured light sensors [12].

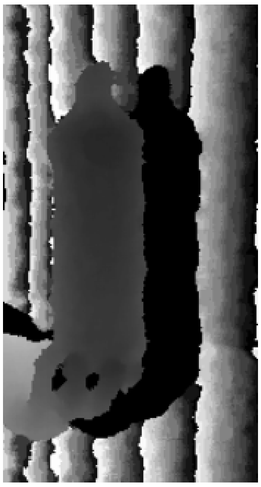


Figure 1.3.3: *A bottle captured using a Kinect camera to illustrate the loss of information in the shadow cast by the object being captured.*

1.3.2 Facial databases

The results from a performance evaluation of a face recognition system is highly dependent on the evaluation data. To enable comparative studies of different systems, the results must therefore be calculated on the same database. To accommodate this, several databases have been compiled using different types of depth data acquisition systems. The databases have subsequently been made freely available for researchers in the field, and employed to run comparable benchmark tests on face recognition systems. A short description of a few selected databases follow and a comparison matrix is tabulated in Table 1.3.1.

Eurecom kinect face dataset

In [14] a database is presented that consists of multi-modal RGBD[3] facial images of 52 people (14 females, 38 males), in 9 different states including facial expressions and partial occlusions, captured using a Kinect [8] sensor, at two different occasions two weeks apart. The dataset includes meta data such as gender, year of birth, ethnicity, eyeglasses and six manually marked facial landmark positions.

The face recognition grand challenge

The Face recognition Grand Challenge (FRGC) [22] was initiated to promote progress in the field of face recognition, and ran from May 2004 to March 2006. The RGBD face image database associated with this project includes 4003 depth images of over 400 subjects, produced using a Minolta Vivid 900/910 series sensor.

The Bosphorus database

To produce images with natural facial expressions, the creators of the Bosphorus database [5] used professional actors as subjects. The database consists of 4666 RGBD face images, taken from different angles, of 105 subjects. Some facial scans include occlusions like beard &

[3] RGBD is an image format that includes both an RGB image and a depth map.

Table 1.3.1: Comparison matrix describing six popular 3D face databases.

Database Properties						
Name	**Sensor**	**Resolution**	**#Subjects**	**Expr.**	**Markers**	**Year**
Eurecom [14]	Kinect	640x480	52	3	6	2012
FRGC [22]	Minolta Vivid 900/910	307/76 Kps	477	1	-	2004
Bosphorus [5]	Inspeck Mega Capturer II 3D	30-50 Kps	105	35	24	2009
UMB-DB [7]	Minolta Vivid VI-900	307/76 Kps	143	4	7	2011
BU-3DFE [28]		1040x1329	100	7	-	2006
BU-4DFE [29]		1040x1329	101	6	-	2008

mustache, hair, a hand or eyeglasses, and up to 35 facial expressions per subject. The images were acquired using a structured light sensor, with a (x,y,z) resolution of (0.3mm, 0.3mm, 0.4mm), and manually marked at 24 facial landmarks.

The university of Milano Bicocca 3D face database

The University of Milano Bicocca 3D face database (UMB-DB) [7] consists of a total of 1473 RGBD images taken of 143 subjects (98 male and 45 female), where a pair of male twins and a baby is included. Four different facial expressions are used and 590 images are partially occluded. The images were created using a Minolta Vivid VI-900 laser depth scanner and annotated with seven facial landmarks. Masks representing the visual part of the face are available for the partially occluded pictures.

Binghamton university 3D facial expression database

In [28] a set consisting of 100 subjects (56 female and 44 male), each performing seven different facial expressions in front of the camera, are presented. The age of the subjects are ranging from 18 to 70 years old.

Binghamton university 3D dynamic facial expression database

With an emphasis on dynamic facial expressions the dataset in [29] are captured as videos with a frame rate of 25 fps. The database consists of 101 subjects, each filmed in six sessions of about 100 frames, doing different facial expressions in different sessions. In total, the set consists of 60600 RGBD frames, each with a resolution of approximately 35000 vertices. The subjects consists of 58 females and 43 males coming from a verity of ethnic backgrounds.

1.3.3 Registration

The problem of face recognition is an intraclass[4] problem, and as such it attempts to classify surfaces that are very similar in shape. To enable the system to detect the small differences in shape, it is advantageous to first register the depth image to a common coordinate system [1, 2, 3, 9, 15, 18, 24, 25, 26]. This is achieved by setting up a cost function that describes how well the data is oriented, and subsequently minimizing this function.

The by far most common approach to this problem, used in [1, 2, 3, 9, 15, 18, 25], is to employ some variant of Iterative Closest Point (ICP). ICP iteratively matches points in the probe image with corresponding points in the reference model and calculates the rigid transform that minimizes the cost function describing their alignment. Using ICP it is possible to relatively quickly align 3D surfaces, however it may get stuck at local optima under certain circumstances. For more details regarding ICP see Section 2.2.

In [24] the cost function was constructed using the Surface Interpenetration Measure (SIM), computed between the probe and the gallery image to be compared, and Simulated Annealing (SA) was subsequently used to minimize this cost function. This approach had the advantage of being less prone to get stuck at local optima during the optimization, but was rather slow compared ICP [24].

A similar approach was taken in [26], using SIM as cost function but a Genetic Algorithm (GA) [21] for the optimization. This provided even better convergence than (SA), but at the expense of being extremely time consuming [24].

Reference model

In much of the early work regarding 3D face recognition, the probe image (i.e the image to be classified) was registered against every gallery image separately. This is called the one-to-all approach. This approach provides very good registration, but at the expense of not scaling well with gallery size. In [15] a reference model was employed, to which both the gallery and probe images was registered, providing a significant speedup without suffering a significant impact on the Recognition Rate (RR) [3]. For more details see Section 2.6.

1.3.4 Feature extraction & classification

When performing classification of a probe image, two approaches are commonly used. Either the probe is compared to all images in the gallery, calculating a one-dimensional similarity measure, or a hyperspace is constructed where probes hopefully will cluster according to the subjects they represents.

Similarity measures

Several different similarity measures have been tested in the context of 3D face recognition. In systems using one-to-all registration, the remaining alignment error is frequently employed (e.g. [24]). The main disadvantage of this approach is that it relies on the one-to-all registration, which is computationally intensive. Another, but related, approach was taken in [25] where an Average Face Model (AFM) was used for registration and the registered surface was

[4] Intraclass, or within class, classification is the problem of distinguishing different instances of the same type of object. In contrast, the problem of general object recognition is an interclass problem.

subsequently resampled on a regular grid to construct a feature vector. This feature vector was projected onto a subspace constructed using Principal Component Analysis (PCA)[5], and classified using a k-Nearest Neighbors (k-NN) classifier. In [26] and [24] SIM was employed, and shown to sometimes produce better results then the remain alignment error, and in [2] Linear Discriminant Analysis (LDA) was used for dimensionality reduction before matching, which led to a significant improvement in RR.

Attacking the problem of 3D face recognition from a different direction, [19] extracted Spherical Harmonics (SH) and used these to train a face classifier, performing well on frontal images.

Facial expression and partial occlusions

The problems of facial expressions and partial occlusions have been tackled from several directions, most having in common that local descriptors have been used. In [1] the face was divided into 5 regions and separately registered using ICP on an AFM, and classified by fusing the separate results discarding regions with high alignment error. A completely different approach was taken in [27] where a 3D adaptation of Scale-Invariant Feature Transform (SIFT), called meshSIFT, was used with good results on a the face database "SHREC 11: face Scans". In [18] and [9] face curves were used as local descriptors, in [18] only as a rejection classifiers to eliminate very dissimilar faces in the gallery, but in [9] for final classification. Facial curves have the advantage of being computationally inexpensive to compute [9], and may be a promising approach.

[5] See Section 2.5

Chapter 2

Theory and method

Face recognition systems are generally the sum of many parts, and the proposed system is no exception. For an intuition regarding how these parts interact see Figure 2.0.1, while a detailed description of the used techniques is the topic in the remainder of this chapter.

2.1 Face orientation using landmarks

The topology of the facial surface is staked out by three-dimensional points, referred to as a point cloud. To enable further processing of the facial surface point cloud ($\mathbf{P^{(S)}}$), it is roughly aligned to a standard orientation. This is accomplished using the positions of the landmarks found on both the aligner face model ($\mathbf{A^{(L)}}$) and probe landmarks ($\mathbf{P^{(L)}}$), by calculating the rigid transform ($\mathbf{R}$ and $\mathbf{T}$) which minimizes the alignment error. The transformation is calculated using Singular Value Decomposition (SVD)[1] as follows:

$$\mathbf{A}_0^{(L)} = \mathbf{A}^{(L)} - \langle \mathbf{A}^{(L)} \rangle \tag{2.1.1}$$

$$\mathbf{P}_0^{(L)} = \mathbf{P}^{(L)} - \langle \mathbf{P}^{(L)} \rangle \tag{2.1.2}$$

$$\text{SVD}(\mathbf{A}_0^{(L)} \mathbf{P}_0^{*(L)}) \implies \mathbf{U \Sigma V}^* \tag{2.1.3}$$

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^*) \end{bmatrix} \mathbf{U}^* \tag{2.1.4}$$

$$\mathbf{T} = \langle \mathbf{A}^{(L)} \rangle - \mathbf{R} \langle \mathbf{P}^{(L)} \rangle \tag{2.1.5}$$

[1] SVD is a matrix factorization technique closely related to eigenvalue decomposition. In fact, the left and right singular vectors ($\mathbf{U}$ and $\mathbf{V}$) are the eigenvectors of $\mathbf{MM}^*$ and $\mathbf{M}^*\mathbf{M}$ respectively, and the singular values (the diagonal entries of $\mathbf{\Sigma}$) is the square roots of the eigenvalues of both $\mathbf{MM}^*$ and $\mathbf{M}^*\mathbf{M}$.

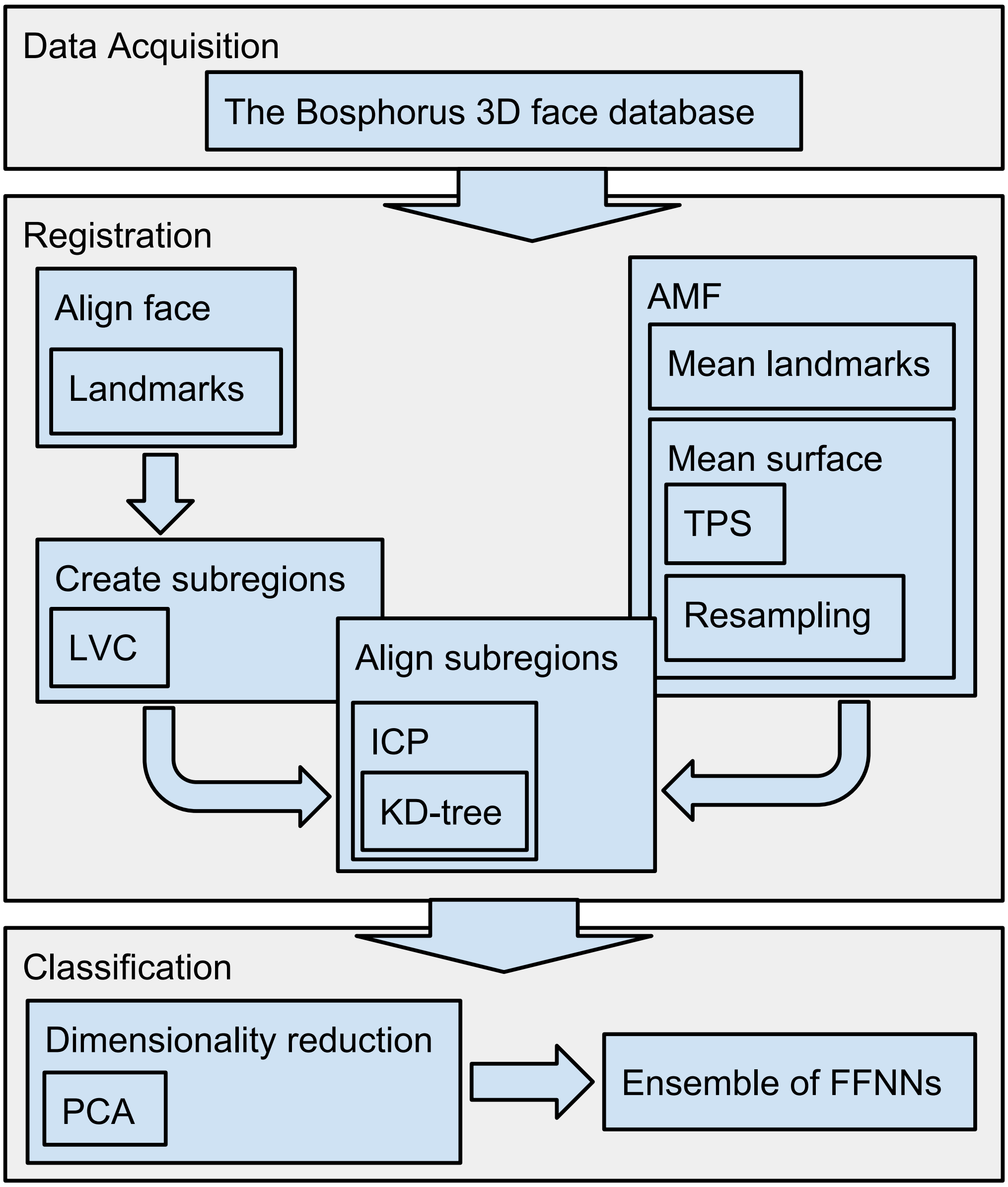


Figure 2.0.1: *Schematic overview of the used techniques and the interacting parts of the system, each described in detail throughout Chapter 2.*

2.2 Iterative closest point

Given two roughly aligned point clouds ($\mathbf{M}$ and $\mathbf{P}$) of N points each, it is possible to efficiently compute the rigid transform which minimizes the point to point distances between them using ICP [6]. ICP works in iterations as described in Algorithm 1, and visualized in Figure 2.2.1.

Algorithm 1: ICP

1. Every point in $\mathbf{P}$ is associated with its closest match in $\mathbf{M}$.
2. The transformation ($\mathbf{R}$ and $\mathbf{T}$) which minimizes the error (e) is calculated.

$$\mathbf{P_0} = \mathbf{P} - \langle\mathbf{P}\rangle \tag{2.2.1}$$

$$\mathbf{M_0} = \mathbf{M} - \langle\mathbf{M}\rangle \tag{2.2.2}$$

$$\text{SVD}(\mathbf{P_0}\mathbf{M_0^*}) \implies \mathbf{U\Sigma V}^* \tag{2.2.3}$$

$$\mathbf{R} = \mathbf{V}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & \det(UV^*)\end{bmatrix}\mathbf{U}^* \tag{2.2.4}$$

$$\mathbf{T} = \langle\mathbf{P}\rangle - \mathbf{R}\langle\mathbf{M}\rangle \tag{2.2.5}$$

3. Transform $\mathbf{P}$ using ($\mathbf{R}$ and $\mathbf{T}$)

$$\mathbf{P} := \mathbf{RP} + \mathbf{T} \tag{2.2.6}$$

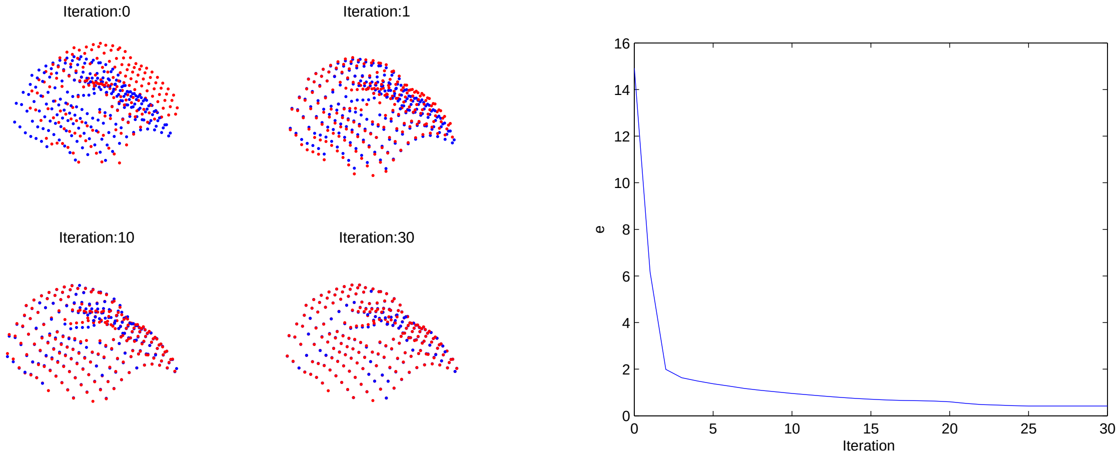
4. Iterate x times or until the error e falls under a given threshold.

$$e = \sqrt{N^{-1}\sum \parallel P - M \parallel} \tag{2.2.7}$$

ICP is proven to find the local minimum by monotonically moving towards lower errors, however it is not guaranteed to find the global minimum which is why the point clouds needs to be roughly aligned before ICP is applied. See Section 2.1 for details regarding rough alignment.

2.2.1 KD-trees

Most of the processing time used during an ICP registration, is spent matching the points between the point clouds. In fact, matching n points in one unstructured point cloud to another, will cost $O(n^2)$ operations. A way to overcome this problem is to structure one of the point clouds in a k-d tree, reducing the cost of n nearest neighbor searches to $O(n \log n)$.



(a) *Point clouds*

(b) *Alinement error*

Figure 2.2.1: *ICP minimizes the point to point distance between two point clouds, here visulaized as red and blue points.*

Nearest neighbor search of a KD-tree

A kd-tree, or k-dimensional tree, is a binary search tree adapted for k-dimensional data. Each node in the search tree splits the data in one of its k dimensions, cycling through the dimensions as the search moves down the tree, see Algorithm 2 for details.

Algorithm 2: Search KD-Tree

1. Starting at the root, move down recursively, going to the left if the search point (P) value of the corresponding dimension is lower than the node value and otherwise to the right.

2. When the algorithm reaches a leaf, this value is stored as the potential nearest neighbor (N_{best}).

3. Now backtrack towards the root, checking each node for:

   (a) If $\| P - N_{current} \| < \| P - N_{best} \|$ than set $N_{best} := N_{current}$.

   (b) Check whether points on the other branch of the current node may be closer. If $(P - N_{current})_{split.dim.} < \| P - N_{best} \|$ then there may be closer points on the other side. This is handled by initiating a new search on this subtree, and saving the result in N_{best} if it is closer than the current best value.

4. When the backtracking completes the root node, the search returns the node N_{best} as the nearest neighbor of the search point P.

Building a KD-tree

Construction of a kd-tree can be accomplished in several different ways, and which particular method that best accommodates the problem at hand depends mostly on how the data is delivered. The canonical method, described in Algorithm 3, can be used when all data is available at build time, and costs $O(n \log n)$ to build.

Algorithm 3: Build KD-Tree ($k = 2$)

1. Find the median value in the full set, for the first dimension. Use the node holding this value as root node and split the rest of the set into two subsets, using the median value as pivot.
2. For each of the two subsets, find the median value of the second dimension, create a node, and split the sets again.
3. Now for each of the four subsets, start over with the first dimension.
4. Continue splitting until all subsets are empty sets, the final nodes becoming the leafs of the tree.

2.3 Thin-plate splines

Thin-Plate Spline (TPS) was introduced in [10] and is based on the physical analogy of the bending energy in a thin sheet of metal, anchored at a fixed set of points. Using TPS a landmarked surface-mesh may be morphed to fit another given set of landmark positions, interpolating the effects on the surrounding surface in a physically plausible way. A point ($\mathbf{x} \in \mathbb{R}^3$) in space is mapped using the input landmarks ($\mathbf{P}$) and the radial basis function:

$$\mathbf{f}(\mathbf{x}; \mathbf{P}, \mathbf{V}) = a_0 + \sum_{k=1}^{3} a_k x_k + \sum_{i=1}^{n} \omega_i \varphi(\|\mathbf{P}_i - \mathbf{x}\|) \tag{2.3.1}$$

The kernel function (φ), that may be chosen to take several different forms, is here defined as

$$\varphi(r) = r \tag{2.3.2}$$

and the parameters ($\mathbf{a}$ and ω) are calculated as

$$[\omega_1, \ldots, \omega_n, a_1, \ldots, a_3]^T = \mathbf{L}^{-1} \left[\begin{array}{c} \mathbf{V}^T \\ \hline \mathbf{0} \end{array} \right] \tag{2.3.3}$$

where $\mathbf{V}$ is the given landmarks towards which the model is warped and $\mathbf{L}$ is calculated as:

$$\mathbf{L} = \left[\begin{array}{c|c} \mathbf{K} & \mathbf{W} \\ \hline \mathbf{W}^T & \mathbf{0} \end{array} \right] \tag{2.3.4}$$

$$\mathbf{W} = [\mathbf{1}|\mathbf{P}] \tag{2.3.5}$$

$$\mathbf{K} = \begin{bmatrix} 0 & \varphi(r_{1,2}) & \ldots & \varphi(r_{1,n}) \\ \varphi(r_{2,1}) & 0 & \ldots & \varphi(r_{2,n}) \\ \ldots & \ldots & \ldots & \ldots \\ \varphi(r_{n,1}) & \varphi(r_{n,2}) & \ldots & 0 \end{bmatrix} \tag{2.3.6}$$

$$r_{i,j} = \|\mathbf{P}_i - \mathbf{P}_j\| \tag{2.3.7}$$

2.4 Resampling

To create comparable fixed dimension surface meshes, the point data will need to be resampled on a uniform grid. This is achieved using linear Delaunay tessellation interpolation [11]. The interpolant is subsequently used to create a mesh on a square grid, scaled to match the resolution of the depth camera.

2.5 Principal component analysis

PCA [16] is a statistical tool used to find the directions in a given dataset, that hold the highest variance. The principal components are the eigenvectors of the covariance matrix of the dataset, sorted according to the associated eigenvalues, in decreasing order. This has the effect of setting the first principal component to the direction in the dataset that has the highest variance, the second to the direction that has the next highest variance, under the constraint that it must be orthogonal to the first principal component. This process continues, in the order of ever decreasing variance, until all dimensions are enumerated. PCA may be implemented using eigenvalue decomposition of the covariance matrix described above, but it is also possible to use SVD to find the principal components directly. The SVD approach is more numerally stable and hence usually recommended.

2.6 Average face model

In a large scale face recognition implementation, it is not feasible to register the probes against every subject in the gallery, in a one-to-all comparison. This is because the registration step, with an order of magnitude, is the most computationally expensive step in the process. A better alternative might be to register all frames in the coordinates of an Average Face Model (AFM), a process which scales as O(1) instead of O(N) and is a simplification that has been shown to have a negligible impact on the RR of the system [25]. Also, using an AFM opens up for the use of machine learning techniques, which as will be later shown can significantly improve the RR of a face recognition system.

The AFM is created in two steps, using a subset of neutral faces from the training set, which will be described below.

2.6.1 Mean landmarks

The mean landmarks are calculated using an iterative process that aims to minimize the difference between the sought reference shape ($\mathbf{A}^{(L)}$) and the mean positions of the landmarks corresponding to all subjects ($\mathbf{P}_i^{(L)}, \forall i \in S$) registered using the reference shape $\mathbf{A}^{(L)}$, see Algorithm 4 for a complete description.

Algorithm 4: Mean Landmarks

```
1. Choose one of the subjects ($i \in S$) as the initial reference (e.g. i=0).
   $\mathbf{A}^{(L)} := \mathbf{P}_0^{(L)}$
2. For each subject ($i \in S$) calculate the rigid transform ($\mathbf{T}_i$ and $\mathbf{R}_i$) that minimizes the Root Mean Squared (RMS) difference between $\mathbf{P}_i^{(L)}$ and $\mathbf{A}^{(L)}$, as described in Section 2.1.
3. Calculate the mean positions for each of the registered landmarks.
   $\mathbf{A}_{next}^{(L)} = \langle \mathbf{R}_i * \mathbf{P}_i^{(L)} + \mathbf{T}_i \rangle, \forall i \in S$
4. Compute the remaining error $\epsilon$ as the RMS difference between the reference shape $\mathbf{A}^{(L)}$ and the mean positions of the landmarks.
   $\epsilon = \mathrm{RMS}(\mathbf{A}^{(L)} - \mathbf{A}_{next}^{(L)})$
5. Assign the mean positions of the registered landmarks to the reference shape.
   $\mathbf{A}^{(L)} := \mathbf{A}_{next}^{(L)}$
6. If the remaining error $\epsilon$ is larger than a set threshold go back to step 2.
7. Normalize the pose of the reference shape $\mathbf{A}^{(L)}$ to use its largest principal component as y-axis, the next biggest as x-axis and its smallest principal component as z-axis. The principal components of the data is found using PCA as described in Section 2.5.
8. Return $\mathbf{A}^{(L)}$
```

2.6.2 Mean facial surface

Using the mean facial landmarks $\mathbf{A}^{(L)}$ the mean facial surface ($\mathbf{A}^{(S)}$) may be constructed as described in Algorithm 5. This surface constitutes the sought AFM, see Figure 2.6.1 for an example.

2.7 Landmark variance clustering

One of the biggest challenges within face recognition stems from the dynamic nature of the face. Facial expressions distort parts of the facial surface, which degrades the performance of classifiers using these distorted regions. A classifier based on the whole face will, hence, only perform well for neutral faces. This can be solved by breaking up the face into smaller regions [1]. However, as can be seen when studying the face recognition performance in humans [17], as well as computers as will be shown later, it is not enough to only analyze the local features of the face, if a high performance system is sought. To combat this issue, the regions are, in the

```
Algorithm 5: Mean Facial Surface
    1. For each subject $i \in S$ calculate the rigid transform ($\mathbf{T}_i$ and $\mathbf{R}_i$) that minimizes the RMS difference between $\mathbf{P}_i^{(L)}$ and $\mathbf{A}^{(L)}$, as described in Section 2.1.
    2. Align the facial surface points ($\mathbf{P}_i^{(S)}$) of each subject $i \in S$ using $\mathbf{T}_i$ and $\mathbf{R}_i$ and warp them using TPS, as described in Section 2.3, to fit the reference landmarks $\mathbf{A}^{(L)}$.
           $\mathbf{P}_i^{(W)} = \text{TPS}(\mathbf{A}^{(L)}, \mathbf{P}_i^{(L)}, R * \mathbf{P}_i^{(S)} + T)$
    3. Resample the surfaces ($\mathbf{P}_i^{(W)}$) on a regular x-y grid, as described in 2.4, to form a set of depth maps $\mathbf{D}_{x,y}^i, \forall i \in S$
           $\mathbf{D}_{x,y}^i = \text{Interpolate}(\mathbf{P}_i^{(W)}, x, y), i \in S$
    4. Average $\mathbf{D}_{x,y}^i$ over all subjects in $S$ and convert back to points and form $\mathbf{A}^{(S)}$
           $\mathbf{A}^{(S)} = \left[x, y, \langle \mathbf{D}_{x,y}^i \rangle\right]$
    5. Return $\mathbf{A}^{(S)}$
```

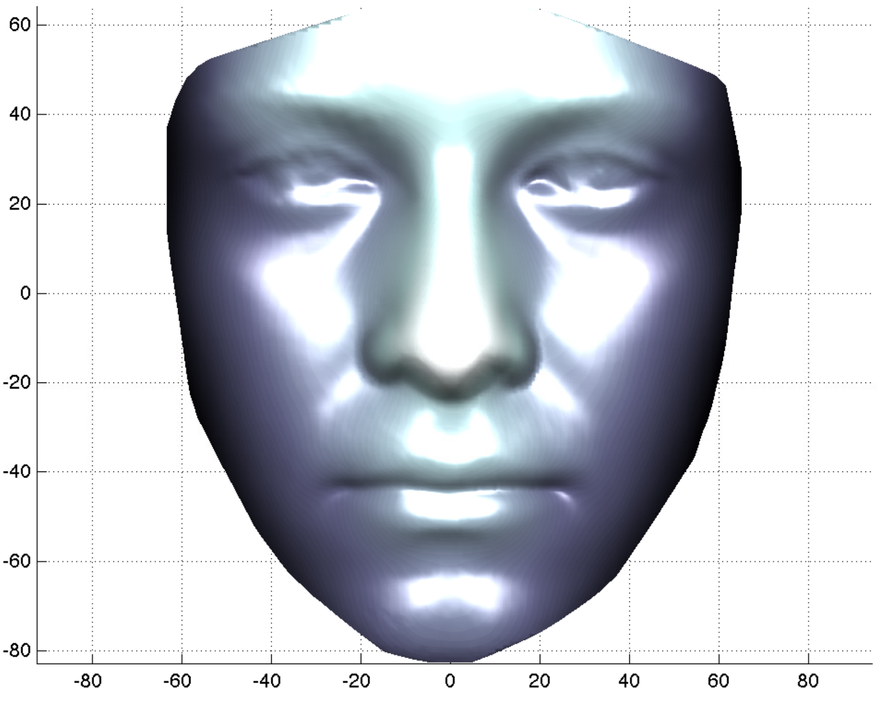


Figure 2.6.1: *AFM constructed from 3D scans of 105 individuals.*

proposed method, grouped to form additional classifiers. However, this leads to the non trivial problem of how to combine these smaller regions to form good classifiers. One approach could be to use all possible combinations of landmarks, however this would yield $\sum_{i=1}^{|L|} \binom{n}{i}$ areas to register and investigate (i.e. the dataset used in this thesis has a cardinality of $|L| = 22$ which equates to 4 194 303 possible combinations of regions), which is computationally infeasible.

In this thesis a novel solution, Landmark Variance Clustering (LVC), to the above problem is presented. LVC aims to find low variance regions, which may be used as discriminative, yet expression invariant, features. LVC achieves this by constructing a landmark Euclidean inter-variance matrix ($\mathbf{d}^{(S)} \in \mathbb{R}^2$) and subsequently performing a hierarchical clustering of this matrix. This approach will yield $2|L| - 1$ regions (i.e. 43 for $|L| = 22$) which is a more manageable number of classifiers for a real time system.

2.7.1 The LVC distance measure

When a face is distorted by facial expressions, the landmarks in the face will move with respect to each other. This interplay may be analyzed to find the clusters of landmarks that are more static with respect to each other, and hence have a greater probability of not being distorted, yet providing a more global perspective of the face.

Using the full training set of subjects ($i \in S$), making all available facial expressions ($e \in E$) the average point to point variance ($d^{(S)}_{l_1,l_2}$) is calculated as:

$$d^{(i)}_{l_1,l_2} = \text{VAR}(\|\mathbf{P}^{(i,e)}_{l_1} - \mathbf{P}^{(i,e)}_{l_2}\|), \forall e \in E \tag{2.7.1}$$

$$d^{(S)}_{l_1,l_2} = \langle d^{(i)}_{l_1,l_2} \rangle, \forall i \in S \tag{2.7.2}$$

where $\mathbf{P}^{(i,e)}$ denotes the facial landmarks of subject i making facial expression e.

2.7.2 Hierarchical clustering

Hierarchical clustering is a technique used to form a hierarchy of clusters from a proximity matrix. The hierarchy is represented by a binary tree structure where the root represents a cluster of all nodes, every branch splitting the set to form two sub clusters, and the leaves representing single nodes, see Figure 2.7.1. The clusters are split using Euclidean distance and single-linkage clustering. The resulting super regions of an LVC analysis can be seen in Figure 3.1.2.

2.8 Dimensionality reduction

When faced with high dimensional data, that is likely to fester dependent variables, it is usually advantageous to try and reduce the dimensionality of the data. Optimally, in such a way that the resulting variables are independent, and with minimal loss of information.

This can be achieved using PCA, see Section 2.5. In this setting, only the first few principal components are kept. The number of dimensions to keep are either given, or decided by discarding the principal components that hold less than a given ratio of the total variance, e.g. components corresponding to less than 0.5% of the variance may be rejected. Finally, using the found principal components, the data is projected onto the new basis.

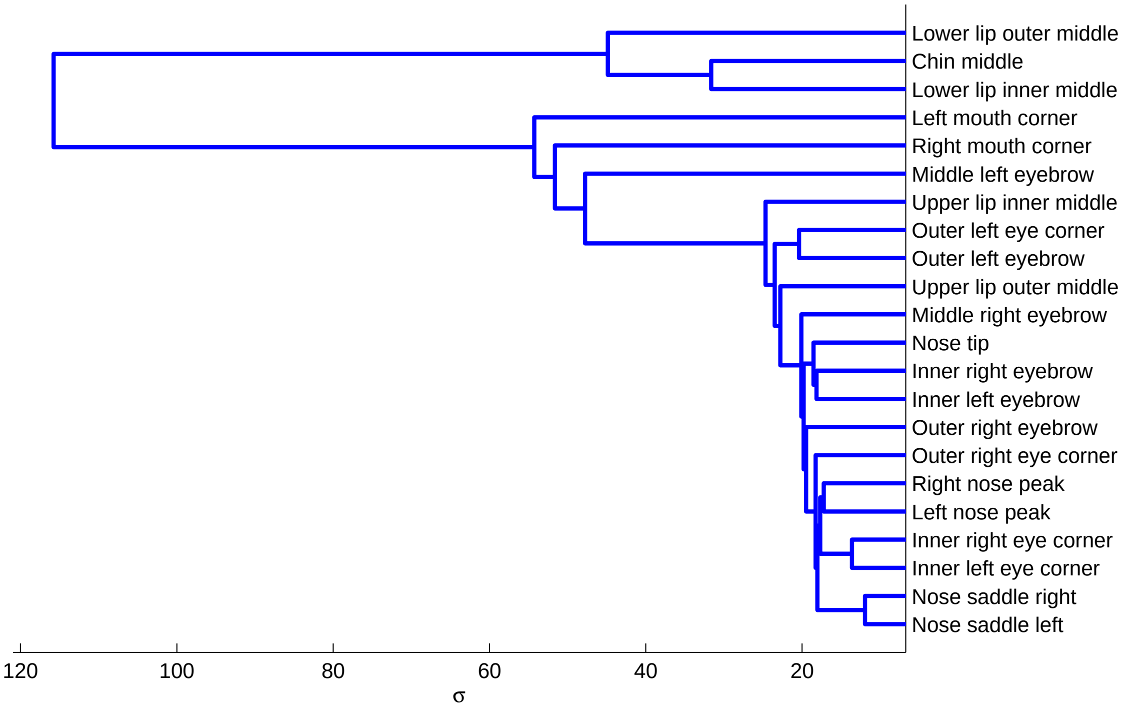


Figure 2.7.1: *Dendrogram representing a hierarchical clustering of facial landmarks, based on Euclidean inter-variance under facial expressions.*

2.9 Artificial neural networks

Artificial Neural Networks (ANN) [13] are biologically motivated and is a machine learning technique that comes in many flavors, see Figure 2.9.1 for an example topology.

2.9.1 Feed-forward neural networks

A Feed-Forward Neural Network (FFNN) [13] is a popular ANN architecture that can be used to perform pattern recognition. The technique is similar to logistic regression using non-linear terms, but does not rely on the user to choose the terms needed to fit the data, making it much more adaptable to complex datasets. The neurons in a FFNN are structured in layers, the first layer called the input layer, the last called the output layer and interim layers are called hidden layers, see Figure 2.9.1. The hidden layers are optional, however, to fit complex patterns they are necessary.

Each neuron is updated as the weighted sum of its inputs, and squeezed into the interval $[0, 1]$ using a sigmoid function.

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2.9.1}$$

The information learned by the neurons is stored in the input weights ($\mathbf{\Theta}$), which are used to calculate the output from each layer ($\mathbf{a}^{(l)}$), eventually producing the network output ($h_{\Theta}(\mathbf{x}) = \mathbf{y}$).

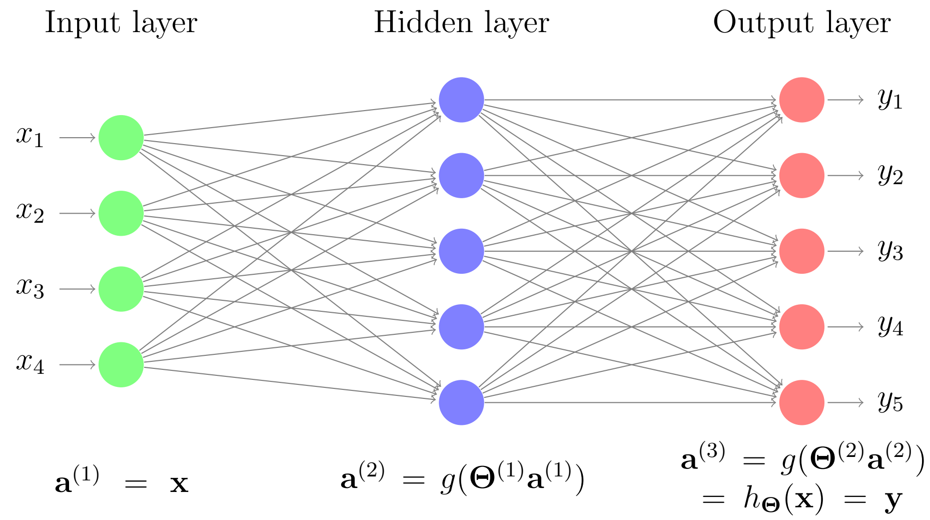

Input layer
Hidden layer
Output layer
x_1
x_2
x_3
x_4
y_1
y_2
y_3
y_4
y_5
$\mathbf{a}^{(1)} = \mathbf{x}$
$\mathbf{a}^{(2)} = g(\boldsymbol{\Theta}^{(1)}\mathbf{a}^{(1)})$
$\mathbf{a}^{(3)} = g(\boldsymbol{\Theta}^{(2)}\mathbf{a}^{(2)}) = h_{\boldsymbol{\Theta}}(\mathbf{x}) = \mathbf{y}$

Figure 2.9.1: *Network diagram representing an example of a FFNN with four input neurons, one hidden layer, and five output neurons. This type of architecture is appropriate when classifying some data* $\mathbf{x} \in \mathbb{R}^4$ *into one of 5 different categories, however depending on the complexity of the input, the number and size of the hidden layers should be scaled accordingly.*

2.9.2 Network training using backpropagation

A FFNN is trained by minimizing the cost function (regularization terms omitted for brevity)

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\Big(-y_k^{(i)}\log((h_\Theta(x^{(i)}))_k) - (1-y_k^{(i)})\log(1-(h_\Theta(x^{(i)}))_k)\Big) \tag{2.9.2}$$

that represent the performance of the network, where m and K denotes the number of input and output neurons respectively.

$J(\Theta)$ is not convex, however it is rather well behaved, and is therefore usually treated as if it were convex when optimized since this opens up for the use of high performance optimization methods, such as the conjugate gradient method. If the algorithm does get stuck at a bad local optimum the optimization is simply restarted. The partial derivatives $\frac{\partial}{\partial\Theta}J(\Theta)$, needed to guide the optimization, are calculated using backpropagation [13].

Chapter 3

Implementation

Two different face recognition approaches are implemented and evaluated in the context of this thesis, see Figure 3.0.1 for a system overview. The first method uses a simple k-NN classifier, implemented to provide a baseline score, and the second uses a FFNN classifier. However, the features used in both systems are identical.

3.1 Feature templates

As a prerequisite for both training and testing, an AFM is constructed as described in Section 2.6.1. The AFM is subsequently divided into 22 basic regions, based on the euclidean distance to the nearest facial landmark, see Figure 3.1.1. In addition to the basic regions, another 21 super regions are constructed based on an LVC, described in Section 2.7, analysis, see Figure 3.1.2. These, in total 43, extracted regions, denoted $\mathbf{C}_i, i \in \{1, \ldots, 43\}$, will later be used as feature templates during feature extraction.

3.2 Registration and feature extraction

Given a probe point cloud ($\mathbf{P}^{(S)}$) of arbitrary orientation with landmarks ($\mathbf{P}^{(L)}$), the first step in the recognition process is rough alignment using landmarks, see Section 2.1. This step is necessary to ensure convergence of the fine grained registration, used during feature extraction.

Next ICP, see Section 2.2, is applied using $\mathbf{P}^{(S)}$ as the model and the first feature template ($\mathbf{C}_1$) as the point cloud to be registered. The reason that $\mathbf{P}^{(S)}$ is used as the model, and not the other way around (which might seem more natural), is that this will implicitly segment the region of interest and not try to match each point in (the much larger) point cloud $\mathbf{P}^{(S)}$ to points in $\mathbf{C}_1$. Finally the registered subregion of $\mathbf{P}^{(S)}$ corresponding to $\mathbf{C}_1$ is re-sampled, see Section 2.4, on the (x,y) grid defined by $\mathbf{C}_1$, and the resulting z values is used to form the feature vector $\mathbf{f}_1$. This processes is repeated for each $\mathbf{C}_i, i \in \{1, \ldots, 43\}$ resulting in the feature vectors $\mathbf{f}_i, i \in \{1, \ldots, 43\}$.

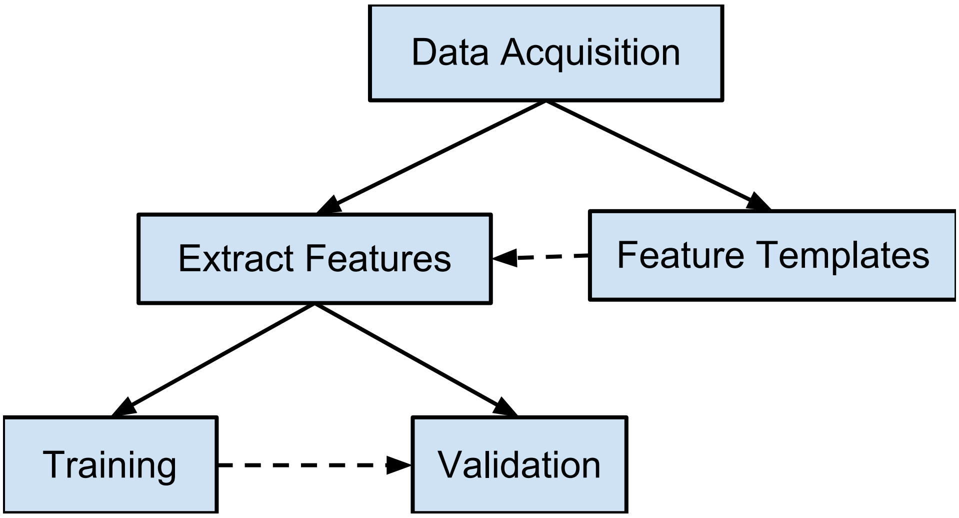


Figure 3.0.1: *System overview depicting the flow of information in the system. The first step, data acquisition, was taken care of when the face database was created, described in Section 1.3.2. Next, using a subset of this data, feature templates are created, see Section 3.1. These are only created ones and are subsequently used in the feature extraction step, Section 3.2, during both training and validation. Training of the classifiers are done first, and subsequently validation of the system is carried out using the learned parameters from the training step, but on a different subset of the data, see Sections 3.3 and 3.4.*

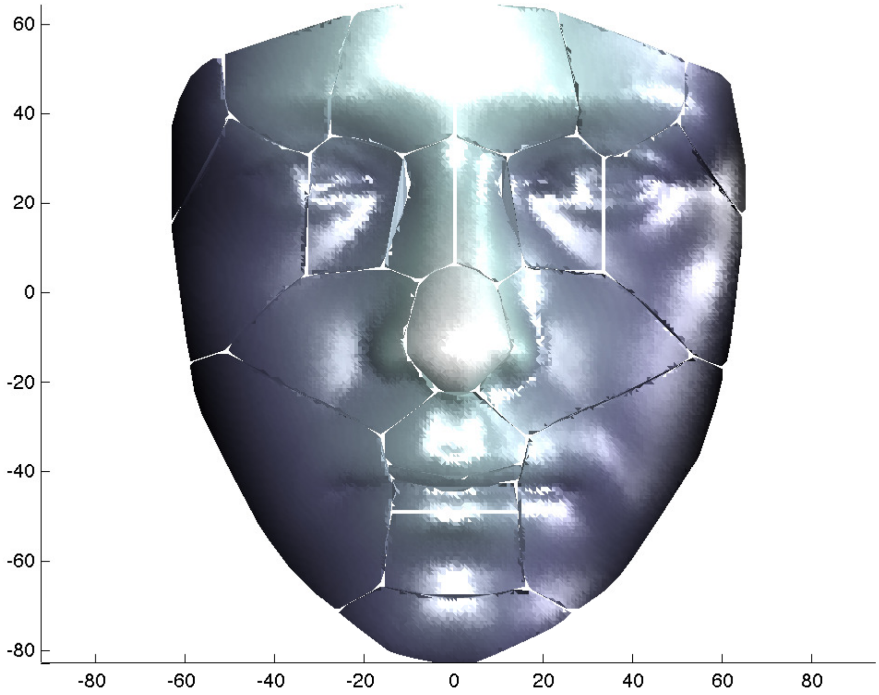


Figure 3.1.1: *An AFM dived into 22 different regions based on the euclidean distance to the nearest facial landmark.*

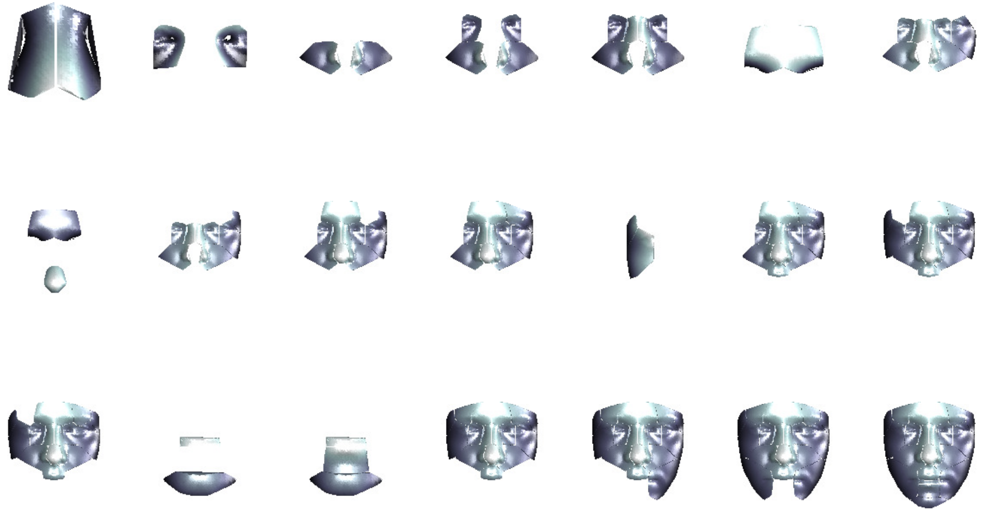


Figure 3.1.2: *21 super regions, constructed using clusters of basic AFM facial regions. The clustering is based on an LVC analysis.*

3.3 k-NN classification

To create a base line performance, k-NN classifiers are trained for each facial region and evaluated.

3.3.1 Gallery generation

Training of the k-NN classifiers are straight forward. It consists of generating feature vectors ($\mathbf{f}_{i,s}, \{i \in \{1, \ldots, 43\}, s \in S\}$) corresponding to one neutral scan of each subject in the data set, and letting these make up the feature space for the classifiers.

3.3.2 Probe classification

For every region, a set of similarity scores are calculated as the inverse of the euclidean distance, in feature space, between the probe and each prototype. The scores are subsequently normalized to zero mean and unit standard deviation withing each region, weighted (excluding regions with an alignment error higher than one standard deviation above the others), and summed to form one score per subject in the gallery. Finally the probe is assigned the identity corresponding to the highest score.

3.4 Neural networks classification

To learn more complicated, non linear, relationships between the feature vectors, a neural network may be used. In this context a set of expert FFNNs, see Section2.9.1, is set up, one for each region, and the results are fused to form the final results.

3.4.1 Preprocessing

Before the feature vectors may be fed to the network, a few preprocessing steps are necessary.

Fix unknowns

Since the face scans may be captured from different angles, there is no guarantee that every point in the feature template has a corresponding point in the probe. In this case that value will be set to the mean value of that point in the training set, and a new feature vector will be generated where the missing value is set to zero and all other to 1.

Normalization

Since the position in space is of no consequence, the feature vectors may without loss of information be normalized to zero mean. Scaling to unit standard deviation may in fact lead to a certain degree of information loss, however it is necessary to be able to do the subsequent sub space projection.

Sub space projection

Training a FFNN using > 20000 input neurons is computationally intensive, and will lead to overfitting if trained using the, in comparison, small dataset used in this thesis. To overcome this problem, a subspace is constructed using PCA, see Section 2.8, into which the feature vectors are projected. The dimensionality of the subspace is set by discarding 0.5% of the total variance, which compresses the feature space to about 30 dimensions ± 5 depending on region.

3.4.2 Network architecture

A pattern recognition FFNN architecture is used for the expert networks, and as such the size of the input layer is defined by the dimensionality of the input vector, and the output layer by the total number of subjects. The number of hidden layers and the number of neurons in each layer are, however, hyper parameters in this model. A few different topologies were tried, among which the setup using three hidden layers with 30, 30, and 60 neurons respectively, gave the best results on the validation set, and is hence the one used in the experiment described in Section 4.4.

3.4.3 Training

Training of a FFNN, as detailed in Section 2.9.2, is a deterministic process. However, it will lead to different results depending on the initial weights, which are randomly chosen. To combat the risk of having one or more of the networks sub-optimally trained, each network is retrained several times, using different initial weights, keeping the network of lowest energy under the constraint that it does not have a negative impact on the validation RR. In practice this means that the algorithm minimizes the validation RR in a greedy[1] best first matter, but allows unchanged validation RR if the new network has lower energy than the current best in that region.

[1] A greedy algorithm uses the heuristic of locally optimal choices, to solve complex problems efficiently.

3.4.4 Sum of expert networks

The output from each expert network is fused using a weighted sum, where the weights is either 0 or 1 depending on the alignment error of the corresponding region. The threshold is set to 0.35 standard deviations above mean alignment error.

Chapter 4

Validation

Validation of the proposed face recognition system is performed on the open benchmarking dataset Bosphorus 3D face database, described in Section 1.3.2. The experiments conducted are designed to measure the performance of the proposed FFNN method and the baseline k-NN method, in both an identification scenario and a verification scenario. The metrics used and the specifics concerning the experiments, are described below.

4.1 Recognition rate

The RR, also denoted Rank 1 RR, measures which portion of the probes that are correctly identified during a trial.

4.1.1 Cumulative match characteristic

A Cumulative Match Characteristic (CMC) curve plots the RR as a function of allowed rank, i.e. the first point in the plot will constitute the rank 1 RR, and the last point will include all ranks, hence it will equal one.

4.2 Verification rate

The Verification Rate is a measure used in verification scenarios. It measures the True Positive Rate (TPR) (i.e. the ratio of frames correctly verified), constrained by a given False Accept Rate (FAR) usually chosen to be 0.1%.

4.2.1 Receiver operating characteristic

The Receiver Operating Characteristic (ROC) curve describes the verification rate, also denoted the true accept rate, as a function of the FAR. If plotted on a linear scale a straight line from the origin to $(1, 1)$ constitutes a random guess system, and in a perfect system the same line will pass through $(0, 1)$. In the face recognition community, the axis corresponding to the FAR is, however, usually plotted on a logarithmic scale.

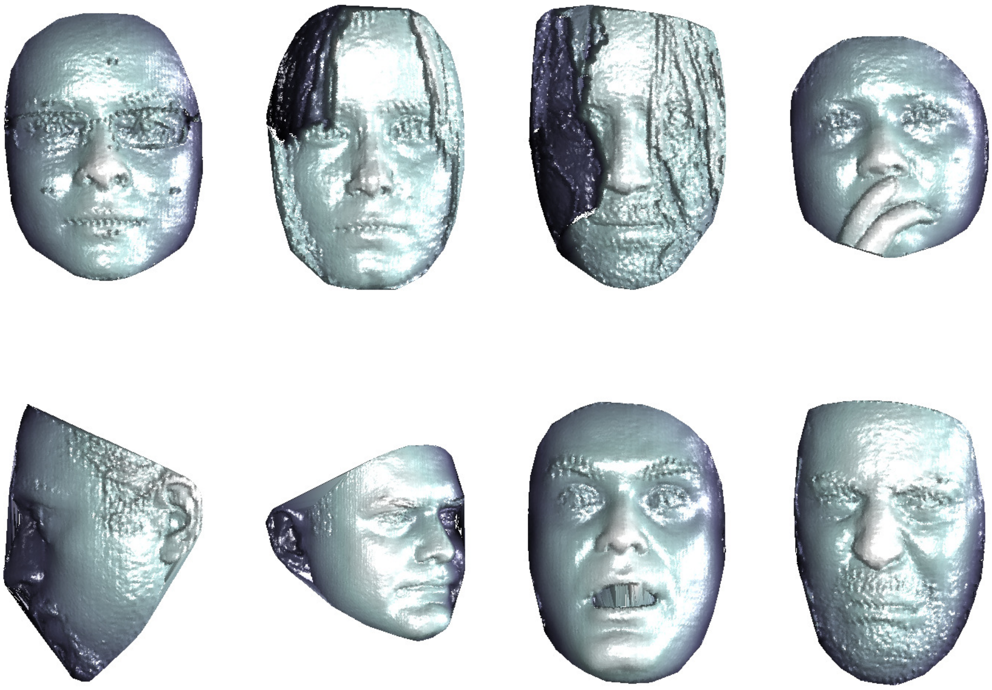


Figure 4.3.1: *Examples of face scans included in the Bosphorus face database.*

4.3 The dataset

Guided by the face database evaluation reported in Section 1.3.2, the Bosphorus face database [5] was selected for evaluation of the system. Bosphorus was selected because it was deemed the most challenging face databases available, exhibiting a wide range of facial expressions, partial occlusions and rotated faces, see Figure 4.3.1.

4.4 Experiments

All performance metrics are calculated from one single experiment per approach. In both cases a classifier is first trained, using the gallery set in the case of k-NN, and the training/validation set in the case of FFNN. Using the trained classifier, a similarity score for each potential identity is calculated for each frame. The resulting output matrix is subsequently used to calculate both identification and validation performance.

Chapter 5

Results

The result of each separate region is discussed in Section 5.1. These results show what discriminate power each separate classifier holds and provides a baseline for the compound classifier discussed in Section 5.2.

5.1 Regional performance

In order to determine the discriminative power of each separate region $\mathbf{C}_i$ a per region RR analysis was conducted. The layout of the basic regions $\mathbf{C}_{\{1,\ldots,22\}}$ can be seen in Figure 3.1.1, and the compound regions $\mathbf{C}_{\{23,\ldots,43\}}$ in Figure 3.1.2. The resulting regional RRs is tabulated in Table 5.1.1.

From this analysis it is apparent that no single region, basic nor compound, can be used to construct an optimal classifier on its own, and that the full facial scan ($\mathbf{C}_{43}$), that was commonly used in early work, constitutes a rather weak classifier when applied to a dataset like the Bosphorus face database.

5.2 Fused results

In order to obtain a stronger classifier, each of the regional scores are fused, as described in Section 3.4.4. The fused scores are further analyzed, as described in Section 4, resulting in the RR and verification rate compiled in Table 5.2.1.

5.2.1 Identification performance

A rank 1 RR of 93.68% was achieved using the baseline k-NN classifier, and a 98.86% RR was achieved by training a FFNN to recognize the feature vectors. The corresponding rank n characteristics, described using CMC curves, can be seen Figure 5.2.1.

5.2.2 Verification performance

In the verification experiment the difference between the two methods were even more evident. The k-NN approach achieving a verification rate of 18.46%, while the approach using a FFNN achieved 95.86%, both measured at 0.1% FAR.

Table 5.1.1: The per region rank 1 RR, as measured on the Bosphorus face database.

Region	Description	k-NN RR(%)	FFNN RR(%)
$\mathbf{C}_1$	Outer left eyebrow	25.66	38.00
$\mathbf{C}_2$	Middle left eyebrow	41.00	51.00
$\mathbf{C}_3$	Inner left eyebrow	51.30	58.57
$\mathbf{C}_4$	Inner right eyebrow	50.84	62.71
$\mathbf{C}_5$	Middle right eyebrow	46.29	43.43
$\mathbf{C}_6$	Outer right eyebrow	37.94	63.29
$\mathbf{C}_7$	Outer left eye corner	51.41	67.43
$\mathbf{C}_8$	Inner left eye corner	37.32	42.43
$\mathbf{C}_9$	Inner right eye corner	46.38	46.71
$\mathbf{C}_{10}$	Outer right eye corner	63.12	68.43
$\mathbf{C}_{11}$	Nose saddle left	31.34	34.86
$\mathbf{C}_{12}$	Nose saddle right	43.07	49.71
$\mathbf{C}_{13}$	Left nose peak	45.90	60.43
$\mathbf{C}_{14}$	Nose tip	48.03	72.71
$\mathbf{C}_{15}$	Right nose peak	55.83	52.14
$\mathbf{C}_{16}$	Left mouth corner	39.41	44.14
$\mathbf{C}_{17}$	Upper lip outer middle	32.33	46.86
$\mathbf{C}_{18}$	Right mouth corner	46.36	54.57
$\mathbf{C}_{19}$	Upper lip inner middle	7.13	1.57
$\mathbf{C}_{20}$	Lower lip inner middle	7.63	6.14
$\mathbf{C}_{21}$	Lower lip outer middle	31.36	39.29
$\mathbf{C}_{22}$	Chin middle	34.74	36.14
$\mathbf{C}_{23}$	$\mathbf{C}_{11} \wedge \mathbf{C}_{12}$	56.40	58.43
$\mathbf{C}_{24}$	$\mathbf{C}_{8} \wedge \mathbf{C}_{9}$	51.97	63.00
$\mathbf{C}_{25}$	$\mathbf{C}_{13} \wedge \mathbf{C}_{15}$	52.49	62.14
$\mathbf{C}_{26}$	$\mathbf{C}_{24} \wedge \mathbf{C}_{25}$	64.20	67.43
$\mathbf{C}_{27}$	$\mathbf{C}_{23} \wedge \mathbf{C}_{26}$	71.03	62.71
$\mathbf{C}_{28}$	$\mathbf{C}_{3} \wedge \mathbf{C}_{4}$	62.75	61.43
$\mathbf{C}_{29}$	$\mathbf{C}_{10} \wedge \mathbf{C}_{27}$	76.16	74.71
$\mathbf{C}_{30}$	$\mathbf{C}_{14} \wedge \mathbf{C}_{28}$	76.85	81.43
$\mathbf{C}_{31}$	$\mathbf{C}_{6} \wedge \mathbf{C}_{29}$	78.55	77.29
$\mathbf{C}_{32}$	$\mathbf{C}_{30} \wedge \mathbf{C}_{31}$	82.80	80.71
$\mathbf{C}_{33}$	$\mathbf{C}_{5} \wedge \mathbf{C}_{32}$	83.08	75.29
$\mathbf{C}_{34}$	$\mathbf{C}_{1} \wedge \mathbf{C}_{7}$	59.74	60.00
$\mathbf{C}_{35}$	$\mathbf{C}_{17} \wedge \mathbf{C}_{33}$	81.97	84.00
$\mathbf{C}_{36}$	$\mathbf{C}_{34} \wedge \mathbf{C}_{35}$	81.12	73.86
$\mathbf{C}_{37}$	$\mathbf{C}_{19} \wedge \mathbf{C}_{36}$	80.46	72.43
$\mathbf{C}_{38}$	$\mathbf{C}_{20} \wedge \mathbf{C}_{22}$	38.35	44.14
$\mathbf{C}_{39}$	$\mathbf{C}_{21} \wedge \mathbf{C}_{38}$	43.64	50.29
$\mathbf{C}_{40}$	$\mathbf{C}_{2} \wedge \mathbf{C}_{37}$	80.64	77.00
$\mathbf{C}_{41}$	$\mathbf{C}_{18} \wedge \mathbf{C}_{40}$	77.86	48.57
$\mathbf{C}_{42}$	$\mathbf{C}_{16} \wedge \mathbf{C}_{41}$	75.12	80.29
$\mathbf{C}_{43}$	$\mathbf{C}_{39} \wedge \mathbf{C}_{42}$	75.05	71.29

Table 5.2.1: The rank 1 RR and verification rate at 0.1% FAR of the purposed methods, as measured on the Bosphorus face database.

Method	Verification Rate(%) at 0.1% FAR	RR(%)
k-NN	18.46	93.68
FFNN	95.86	98.86

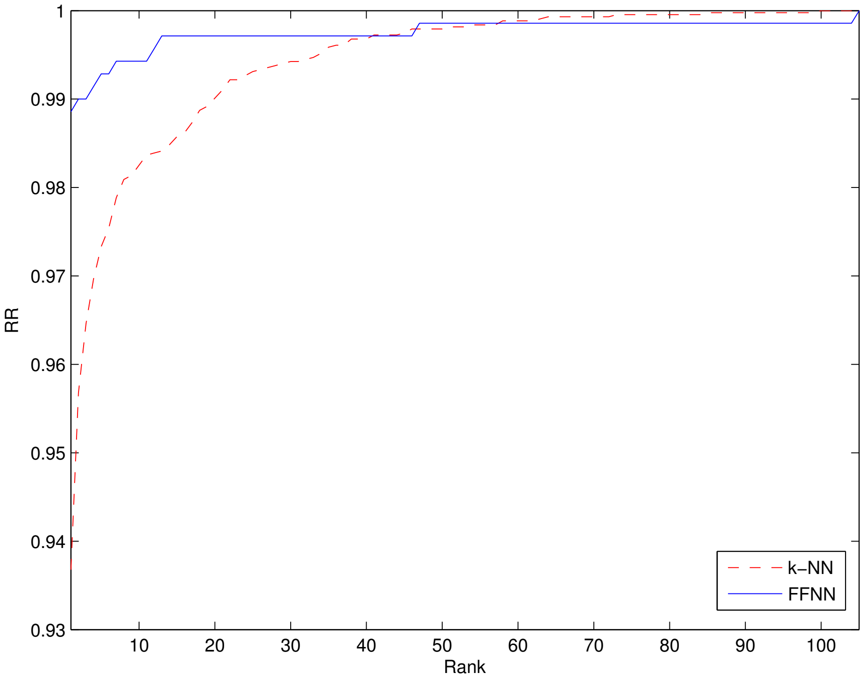


Figure 5.2.1: *The CMC curves corresponding to the proposed method and the baseline method, as measured on the Bosphorus face database. Both systems show similar characteristics, but the FFNN approach is much better in the top ranks.*

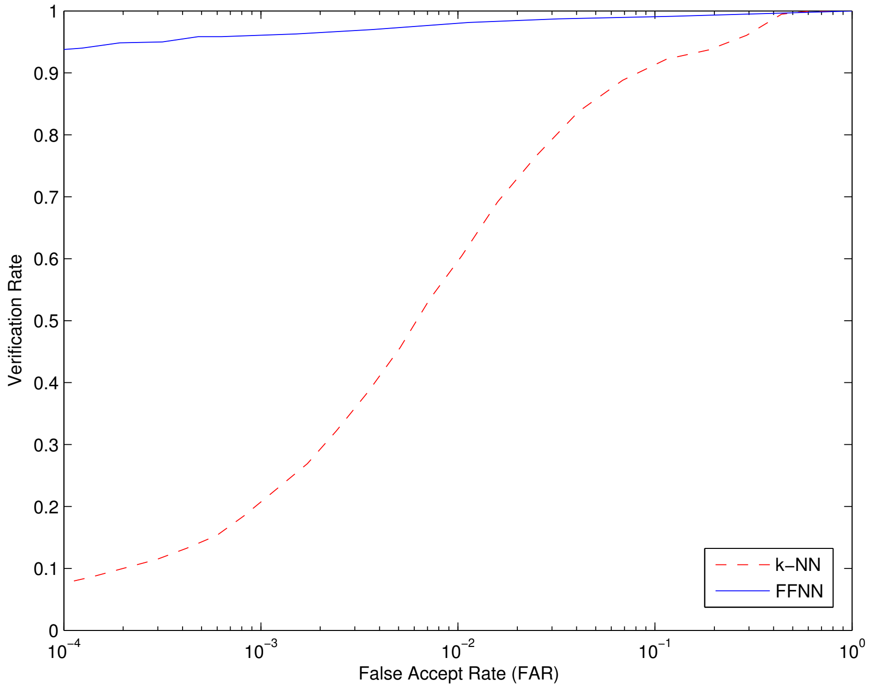


Figure 5.2.2: *The ROC curves of the proposed method and the baseline method, as measured on the Bosphorus face database. The FFNN classifier achieves much greater security than the simple k-NN approach. The FFNN approach verifying the identity of over* 90% *of the probes when accepting a risk of less then* 0.01% *of false verification of an intruder, compared to about* 10% *FAR for the k-NN classifier at the same verification rate.*

The ROC curves, describing the systems verification rate as a function of FAR, is illustrated in Figure 5.2.2.

Chapter 6

Discussion and conclusions

A novel 3D face recognition approach is presented, outperforming all existing methods found in current research. The method uses local and global features, extracted using the novel landmark variance clustering method LVC and an AFM for registration. Classification is done using FFNNs, achieving a top rank 1 RR score of 98.86% and a verification rate of 95.86% at 0.1% FAR, on the challenging dataset the Bosphorus face database.

By choosing the Bosphorus face database for verification, the success conditions stipulated in Section 1.1 could be verified, and considering the results achieved, the conditions can be considered met.

A short description of the main contributions of this thesis follows.

6.1 Rough alignment

To handle faces rotated up to 90°, a robust rough alignment method had to be developed. The purposed method, described in Section 2.1, uses all 22 available landmarks on the aligner face, and is able to converge as long as at least three landmarks can be found on the probe face, no matter which of the landmarks this may be. However, if more landmarks are found on the probe, this will lead to a less noisy result, less dependent on exact landmarking.

6.2 LVC

The novel LVC method, described in Section 3.1.2, was developed to enable the algorithm to choose which parts of the face that are more likely to be rigid. This information is used by the system to create larger facial areas, staked by several facial landmarks, in a way that minimizes the distortion introduced by facial animations.

Table 6.4.1: Rank 1 RR of the purposed methods compared to state of the art results, measured on the Bosphorus face database.

Method	RR(%)	Subjects	Note
k-NN [this]	93.68	105	
MeshSIFT [20]	93.7	105	
ARM [1]	95.29	47	
Spherical Harmonics [24]	95.63	105	90° scans excluded
AvRM+LDA [2]	98.19	105	Only frontal and non-occluded scans
FFNN [this]	**98.86**	105	

6.3 Classification using FFNN

To enable the use of machine learning methods, fixed length feature vectors were generated using AFM registration and regular re-sampling. These were subsequently classified using a FFNN, significantly outperforming the reference k-NN classifier. Concretely, the RR was improved from 95.86% to 98.86% and the 0.1% FAR verification rate from 18.46% to 93.68%.

6.4 Performance comparison

During the literature study in conjunction with this thesis, state of the art results on the Bosphorus database were compiled, see Table 6.4.1. The metric most commonly reported in the literature is the RR, and therefore, this metric was selected as the benchmark score. From the compiled list it is apparent that the result achieved using the proposed FFNN approach is significantly improving the state of the art, considering that the results close behind the FFNN approach are achieved on a less challenging subset of the database. The best found result on the full database was 93.7% which means that the achieved result in this thesis is an improvement by 5.2%.

Chapter 7

Future work

7.1 Real time implementation

The prototype was implemented using Matlab, achieving an acceptable recognition time of a few seconds, the majority of which was spent during registration. However, if the system is to be used in a real time application, e.g. a humanoid robot, a frame rate of about 20 fps will likely be needed. To achieve this, one solution might be to use a GPU implementation of ICP.

7.2 On-line learning

A drawback with using a FFNN, is that it cannot be continually updated with new information. In fact, if new data is to be incorporated without gradually loosing the already learned, the network will need to be retrained using all previously learned data in addition to the new data. This type of learning is refereed to as off-line learning, and is clearly not a plausible solution if a continually learning system is required. However, there are other learning methods, e.g. incremental learning vector quantization, that are able to handle on-line learning, and that could be used instead of the FFNN to create a classifier that is able to learn as it goes along.

7.3 Automatic landmarking

To ease up the face recognition problem, manually landmarked face scans were used. In a realistic scenario this landmarking would, however, necessarily need to be automatically accomplished. Automatic landmarking is an active area of research, but it would still be interesting to see how well the proposed system would perform using current, state of the art, automatic landmarking.

7.4 Advanced machine learning classifiers

Using FFNNs to classify the probe images proved a big step forward in system performance. There are, however, many more advanced machine learning algorithms that may provide even better results. It might for instance be rewarding to apply the deep learning paradigm, and learn the mapping between the expert FFNNs and the final output.

7.5 Deploy in a social robot

The ultimate challenge for a face recognition system is to learn, not only the difference between different subject, but the intricate facial interplay used by humans in social interactions. The localized features employed by the proposed approach make it a plausible candidate for such a system.

References

[1] N. Alyuz, B. Gokberk, and L. Akarun. “A 3D Face Recognition System for Expression and Occlusion Invariance”. In: *2008. BTAS 2008. 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*. Sept. 2008.

[2] N. Alyuz, B. Gokberk, and L. Akarun. “Regional Registration for Expression Resistant 3D Face Recognition”. In: *IEEE Transactions Information Forensics and Security* (Sept. 2010).

[3] N. Alyuz et al. “3D Face Recognition Benchmarks on the Bosphorus Database with Focus on Facial Expressions”. In: *Biometrics and Identity Management*. 2008.

[4] Binghamton University. *Analyzing Facial Expressions in Three Dimensional Space*. Binghamton University. Oct. 2008. URL: `http://www.cs.binghamton.edu/~lijun/Research/3DFE/3DFE_Analysis.html`.

[5] Bogazici University. *The Bosphorus Database*. Bogazici University. 2009. URL: `http://bosphorus.ee.boun.edu.tr/default.aspx`.

[6] Y. Chen and G. Medioni. “Object Modeling by Registration of Multiple Range Images”. In: *1991 IEEE International Conference on Robotics and Automation*. Apr. 1991.

[7] A. Colombo, C. Cusano, and R. Schettini. *The University of Milano Bicocca 3D face database*. University of Milano Bicocca. 2011. URL: `http://www.ivl.disco.unimib.it/umbdb/index.html`.

[8] C. Dal Mutto et al. “CW Matricial Time-of-Flight Range Cameras”. In: *Time-of-Flight Cameras and Microsoft Kinect^{TM}*. Springer US, 2012.

[9] H. Drira et al. “3D Face Recognition Under Expressions, Occlusions and Pose Variations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Feb. 2013).

[10] J. Duchon. “Splines minimizing rotation-invariant semi-norms in Sobolev spaces”. In: *Constructive Theory of Functions of Several Variables*. 1976.

[11] L. D. Floriani, B. Falcidieno, and C. Pienovi. “Delaunay-based representation of surfaces defined over arbitrarily shaped domains”. In: *Computer Vision, Graphics, and Image Processing* 32.1 (1985), pp. 127–140. ISSN: 0734-189X. DOI: `http://dx.doi.org/10.1016/0734-189X(85)90005-2`. URL: `http://www.sciencedirect.com/science/article/pii/0734189X85900052`.

[12] S. Foix, G. Alenyà, and C. Torras. “Lock-in Time-of-Flight (ToF) Cameras: A Survey”. In: *IEEE Sensors Juornal* (2011).

[13] S. Haykin. *Neural Networks: A Comprehensive Foundation*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994. ISBN: 0023527617.

[14] T. Huynh, R. Min, and J.-L. Dugelay. “An Efficient LBP-based Descriptor for Facial Depth Images applied to Gender Recognition using RGB-D Face Data”. In: *ACCV Workshop on Computer Vision with Local Binary Pattern Variants*. Nov. 2012.

[15] M. Irfanoglu, B. Gokberk, and L. Akarun. “3D Shape–based Face Recognition using Automatically Registered Facial Surfaces”. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Aug. 2004.

[16] I. Jolliffe. “Principal Component Analysis”. In: *Encyclopedia of Statistics in Behavioral Science.* John Wiley & Sons, Ltd, 2005. ISBN: 9780470013199. DOI: 10.1002/0470013192.bsa501. URL: http://dx.doi.org/10.1002/0470013192.bsa501.

[17] N. Kanwisher, J. McDermott, and M. M. Chun. “The fusiform face area: a module in human extrastriate cortex specialized for face perception”. In: *The Journal of Neuroscience* (1997).

[18] X. Li and F. Da. “Efficient 3D face recognition handling facial expression and hair occlusion”. In: *Image and Vision Computing* (Sept. 2012).

[19] P. Liu et al. “Learning the Spherical Harmonic Features for 3-D Face Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Mar. 2013).

[20] C. Maes et al. “Feature detection on 3D face surfaces for pose normalisation and recognition”. In: *2010 Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS).* Sept. 2010.

[21] K. F. Man, K. S. Tang, and S. Kwong. “Genetic algorithms:Concepts and aplications”. In: *IEEE Transactions on Industrial Electronics.* Oct. 1996.

[22] J. Phillips. *Face Recognition Grand Challenge.* The National Institute of Standards and Technology (NIST). Feb. 2011. URL: http://www.nist.gov/itl/iad/ig/frgc.cfm.

[23] P. J. Phillips et al. “Symmetric surface-feature based 3D face recognition for partial data”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence.* May 2010.

[24] C. C. Queirolo et al. “3D Face Recognition Using Simulated Annealing and the Surface Interpenetration Measure”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Feb. 2010).

[25] A. A. Salah, N. Alyuz, and L. Akarun. “Registration of three-dimensional face scans with average face models”. In: *Journal of Electronic Imaging* (2008).

[26] L. Silva et al. “Range image registration using enhanced genetic algorithms”. In: *2003 International Conference on Image Processing, 2003. ICIP 2003. Proceedings.* Sept. 2003.

[27] D. Smeets et al. “Symmetric surface-feature based 3D face recognition for partial data”. In: *2011 International Joint Conference on Biometrics (IJCB).* Oct. 2011.

[28] L. Yin et al. “A 3D Facial Expression Database For Facial Behavior Research”. In: *7th International Conference on Automatic Face and Gesture Recognition (FGR06).* Apr. 2006.

[29] L. Yin et al. “A High-Resolution 3D Dynamic Facial Expression Database”. In: *The 8th International Conference on Automatic Face and Gesture Recognition (FGR08).* 66. Sept. 2008.

[30] J. Zhu et al. “Fusion of time-of-flight depth and stereo for high accuracy depth maps”. In: *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.* June 2008.