Karlsruhe Institute of Technology

Facial Image-based Age Estimation

Study Thesis of

Matthias Steiner

At the faculty of Computer Science
Institute for Anthropomatics
Facial Image Processing and Analysis

Advisor: Dr. Hazım Kemal Ekenel

Time: 01. Juli. 2010 – 30. September 2010

Abstract

This work presents a machine learning approach for facial image-based age estimation. The idea is to first extract age relevant texture and shape features from a set of images and use them in combination with the subject's age to learn a model of the human aging process. The classification is done in two steps. At first a classification between youths and adults is done. In the second step the exact age is estimated by a more specific classifier based on the result of the first step. Extensive experiments on the FG-NET aging database are conducted using the leave one person out evaluation scheme. Modifications regarding the feature extraction and taking a soft decision in the first step of the classification are found to improve the performance, leading to a mean absolute error of 4.77 years, which is the lowest mean absolute error reported on the FG-NET aging database.

Contents

1. Introduction **7**
- 1.1. Motivation . . . 7
- 1.2. Related Work . . . 7
- 1.3. Aging Databases . . . 11
- 1.4. Approach . . . 12
- 1.5. Outline . . . 12

2. Methodology **13**
- 2.1. Active Appearance Models . . . 13
- 2.2. Support Vector Machine . . . 14
  - 2.2.1. The Kernel Trick . . . 15
  - 2.2.2. Soft Margin . . . 16
  - 2.2.3. Regression (ϵ-SVR) . . . 16
  - 2.2.4. Multiclass Classification . . . 17

3. Implementation **18**
- 3.1. AAM Building . . . 18
- 3.2. AAM Fitting . . . 18
- 3.3. Age Estimation . . . 21

4. Experiments **23**
- 4.1. Evaluation . . . 23
  - 4.1.1. Comparison with Khoa Luu et al. work . . . 23
  - 4.1.2. Leave One Person Out . . . 25
- 4.2. Modifications . . . 31
  - 4.2.1. Using DCT Features . . . 31
  - 4.2.2. Global Classifier Modification . . . 33
  - 4.2.3. Soft Youth/Adult Classification . . . 33
- 4.3. Comparison to Other Estimation Techniques . . . 35
- 4.4. Automatic Initialized Fitting . . . 37
  - 4.4.1. Evaluation . . . 38

5. Conclusion **41**
- 5.1. Future Work . . . 41

A. Appendix **48**

List of Figures

1.1. Pressure behavior model of a fluid-filled spherical object 8
1.2. Age estimation steps 9
1.3. Some images of the FG-NET database with landmarks 11

2.1. Triangles used for warping (left) and shape free image (right) 14
2.2. Example for an optimal hyperplane 14
2.3. Example for the transformation into a higher dimensional space 16

3.1. Variation of the first five combined features in both directions 19
3.2. Some fitting results, AAM trained on the FG-NET database 20
3.3. The age classifier 21

4.1. Age distribution of the test and training set of Khoa Luu et al. work . . . 24
4.2. Cumulative score comparison for b_s, b_t, b_s concatenated with b_t and b_c, tested with the LOPO method on the FG-NET database 28
4.3. Some youth/adult classification errors, tested with the LOPO method on the FG-NET database, using the combined feature vector 29
4.4. Some large youth age estimation errors, tested with the LOPO method on the FG-NET database, using the combined feature vector 30
4.5. Some large adult age estimation errors, tested with the LOPO method on the FG-NET database, using the combined feature vector 30
4.6. Examples for the image alignment 31
4.7. Cumulative score comparison for b_c, b_{d_1}, b_{d_2} and b_{d_2} concatenated with b_s, tested with the LOPO method on the FG-NET database 32
4.8. The age classifier with the soft youth/adult classification 34
4.9. Cumulative score comparison with and without soft youth/adult classification for b_c and b_{d_2} concatenated with b_s, tested with the LOPO method on the FG-NET database 35
4.10. Cumulative scores comparison to other age estimation techniques, tested with the LOPO method on the FG-NET database 37
4.11. Example for the automatic initialized AAM fitting procedure. 1. MCT face and eye detection (left), 2. initial shape (middle), 3. converged shape (right) 38
4.12. Cumulative score comparison for some feature combinations with and without automatic initialized AAM fitting 40

List of Tables

4.1. MAE finding of our implementation on the training and testing set used in Khoa Luu et al. work 25
4.2. MAE for b_s, b_t, b_s concatenated with b_t and b_c, tested with the LOPO method on the FG-NET database 26
4.3. Age ranges MAE for b_s, b_t, b_s concatenated with b_t and b_c, tested with the LOPO method on the FG-NET database 27
4.4. MAE for b_c, b_{d_1}, b_{d_2} and b_{d_2} concatenated with b_s, tested with the LOPO method on the FG-NET database 32
4.5. Age ranges MAE for b_c and concatenation of b_{d_2} and b_s, tested with the LOPO method on the FG-NET database 33
4.6. Age ranges MAE for b_c and concatenation of b_{d_2} and b_s with soft youth/adult classification, tested with the LOPO method on the FG-NET database . . 34
4.7. MAE comparison to other age estimation techniques, using LOPO method on the FG-NET database 36
4.8. MAE comparison for all feature combinations with and without automatic initialized AAM fitting 39

Abbreviations

AAM **A**ctive **A**ppearance **M**odel

AGES **Ag**ing patt**e**rn **S**ubspace

ASM **A**ctive **S**hape **M**odel

CS **C**umulative **S**core

FGnet **F**ace and **G**esture Recognition Research **Net**work

LAR **L**east **A**ngle **R**egression

LDA **L**inear **D**iscriminant **A**nalysis

LOPO **L**eave **O**ne **P**erson **O**ut

MAE **M**ean **A**bsolute **E**rror

PCA **P**rincipal **C**omponent **A**nalysis

SVC **S**upport **V**ector **C**lassification

SVM **S**upport **V**ector **M**achine

SVR **S**upport **V**ector **R**egression

1. Introduction

1.1. Motivation

In the area of computerized analysis of facial images for recognition, ethnicity classification, gender recognition, etc. the age estimation is a barely explored part. However recently the interest in this subject has significantly increased, because it has many practical applications. For example there are age limitations for driving a car, buying alcohol, cigarettes, films, video games, etc. which should be obeyed, but the human skills of age estimation are very limited. So a computer system, which supports the responsible persons would be helpful. It is well known that the human-computer interaction varies for different age groups, thus a system which automatically adapts its interface to the age of the current user would clear this problem. The knowledge about human age could also be used to even improve other areas of computational image analysis like face recognition, by simulating the aging process for outdated images.

The objective of this work was to get an overview of the current methods and to develop our own solution based on these ideas. The approach of Khoa Luu et al. [29] has been benefited during the study and became the starting point of this work.

1.2. Related Work

Caused by the growing interest a lot of papers have been published over the last years, which deal with the problem of age estimation. They outline some different and interesting approaches.

The work of Narayanan Ramanathan et al. [38] provides a good introduction to the topic. They examine the problem from a more wide point of view, namely the analysis of the basics of human face aging and what has been done there so far. The first steps in understanding the morphological changes associated with growth in biological forms were made by D'arcy Thompsons's study of morphogenesis (1917) [47]. Based on Thompson's work, Shaw et al. [43] studied facial growth as an event perception problem and discovered the *Cardioidal strain* and the *Affine Shear* transformation to describe facial growth. Further Pittenger and Shaw [35] enhanced this approach and distinguish the importance of three force configurations, named: the *shear*, *strain* and *radial* forces and identified the *cardioidal* strain forces (containing the radial components) to be the most important one. Todd et al. [50] developed the *'revised' cardioidal strain* transformation model, by comparing the human head growth with the modeling of a fluid-filled spherical object with pressure, see Figure 1.1. Finally Mark et al. [30] hypothesized that the information associated with any recognizable style change is contained in geo-

metric invariants associated with the event. In reference to facial growth they identified three geometric invariants, whose attributes have to be preserved across transformation. Narayanan Ramanathan et al. summarized that in computer vision the facial aging problem can be categorized in *Shape vs. Texture*, *Feature selection* and *Factors*. After this they discussed a couple of age estimation techniques, like the classification based on anthropometry of the face by Kwon and Lobo [21], the Active Appearance Model (AAM) by Lanitis et al. [25] [23] and the AGES method working with so called aging pattern by Geng et al. [17]. In the next chapter they listed a few computational model for age progression, developed by Burt and Perrets's [5], Tiddeman et al. [49] [48], Ramanathan and Chellappa [36] [37] and Suo et al. [45]. Finally they gave a survey about existing aging databases, which are: MORPH [1] [39], FG-NET [13] and FERET [34].

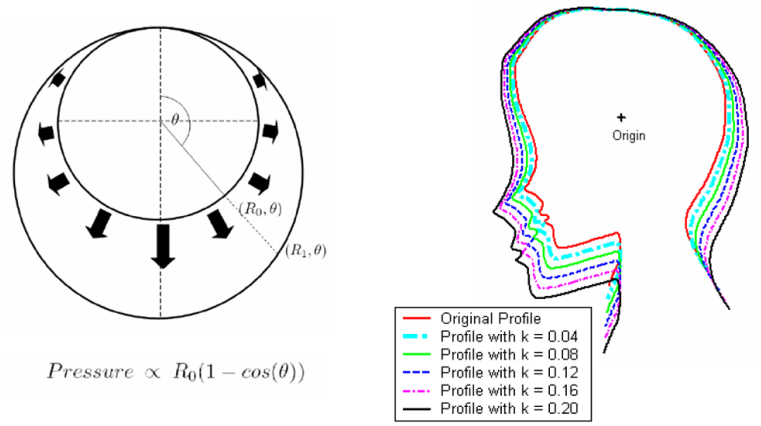


Figure 1.1.: Pressure behavior model of a fluid-filled spherical object left and the resulting *'revised' cardioidal strain* model right [50].

Xin Geng et al. [17] developed an age estimation method named *AGES* (AGing pattErn Subspace), based on the following assumptions:

1. The aging progress is uncontrollable.
2. Every person ages differently.
3. The aging progress must obey the order of time.

Therefore they introduced the so called aging patterns, as a sequence of personal facial images sorted in chronological order. The images are represented by their feature vector, extracted by the Appearance Model described in [10]. Instead of using isolated pictures for training, a subspace of the aging patterns is learned, using Principal Component Analysis (PCA). A big problem was the lack of complete aging patterns, which led to highly incomplete training data. To deal with this they developed an iterative learning algorithm, which is able to estimate a part of the missing personal aging pattern with each iteration, using the global aging pattern model learned so far. The updated personal

model is then used to refine the global one. When estimating the age, the feature vector of the image is calculated and used to find a suitable aging pattern in the subspace. This is done, by searching for the projection in the subspace achieving the smallest reconstruction error of the feature vector, which is illustrated in Figure 1.2. In the second step the proper position in the determined aging pattern is indicated by the minimal reconstruction error.

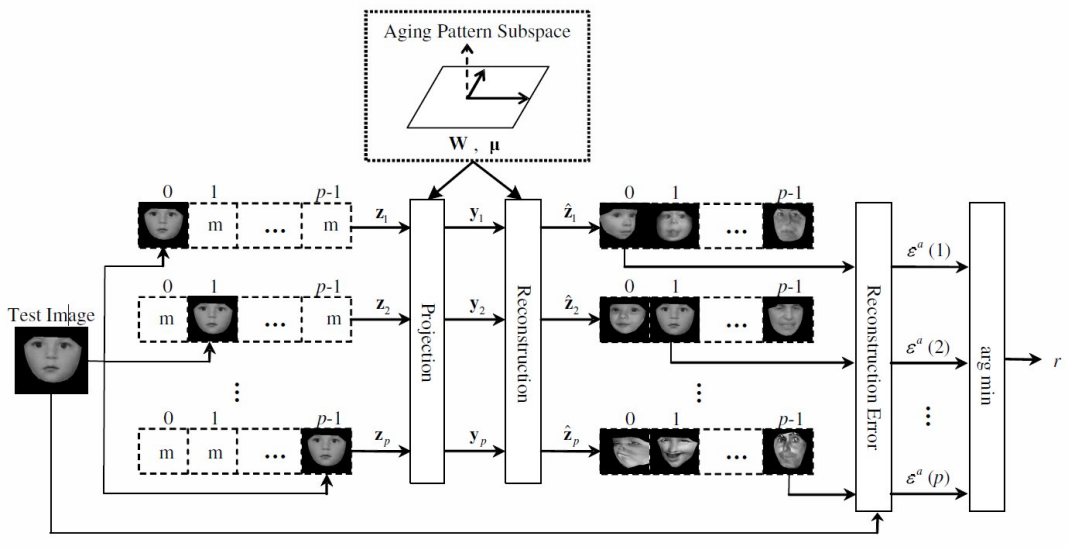


Figure 1.2.: Age estimation steps, missing parts are marked by 'm' [17].

Based on their earlier work [17], Xin Geng and Zhi-Hua Zhou [18] introduced an improved version of *AGES*, named $AGES_{LDA}$. They additionally applied the Linear Discriminant Analysis (LDA) to the feature vectors extracted by the Appearance Model, to deal with expression variations, pose and illumination. They also built a two-layer age estimation, by first using *AGES* to classify the test samples into the three most consistent age ranges. Then they used three separately trained subspaces to assign the exact age.

Karl Ricanek et al. [40] used the Active Appearance Model (AAM) described in [46], to locate relevant aging features. In the next step they applied Least Angle Regression (LAR) by Efron et al. [11] to identify the most important features. The reduced feature vectors of the training set are then used for Support Vector Regression (SVR) by Vapnik [51]. While training, the age estimation acts as a feedback for the LAR. They also tried to incorporate information like race and gender into their training, but got the best performance, when only using the results of the LAR.

Feng Gao and Haizhou Ai [14] collected thousands of frontal or near frontal face images and labeled them with a subjective age. The training samples were divided into the following four age groups: 0-1, 2-16, 17-50 and 50+. To get a descriptor vector they used Gabor features by C. Liu et al. [26]. Then they utilized the linear discriminant analysis (LDA) technique [4], to build the classifier. They further improved it by implementing

a Fuzzy LDA version, to cope with the fact that the classification into the age groups is too strict in the border areas. For evaluation they also tried different features like LBP and other classification methods like SVM, arriving at the conclusion that Gabor features combined with LDA is the best choice.

Khoa Luu et al. [29] used Active Appearance Model (AAM) to extract a combined feature vector of the facial images. The classifier is divided into two main steps. First a binary classifier f is build by SVMs to distinguish between youths (0-20) and adults (21-69). In the second step a growth and development function f_1 and an adult aging function f_2 are separately trained with Support Vector Regression (SVR) [27], on youth and adult datasets, respectively. When classifying, the test image is first assigned to one of the two age groups and then handed to the corresponding age function, to estimate the exact age. Based on this work Khoa Luu et al. modified the classifier construction by adding a supervised spectral regression after the extraction of the combined AAM feature vector [28]. It should improve the correlation information among the feature vectors of the same class and decrease it for different classes. Also it should help to reduce the dimension of the feature vector.

Based on a recent work, Sethuram et al. [41] improved their analysis-synthesis face-model approach, which is based on *Active Appearance Models* (AAM). They used *Support Vector Regression* (SVR) to learn age-based properties of the AAM parameters and gradient-regression-based AAMs, to represent the texture information. After this a Monte-Carlo simulation is run, which generates random faces, which are then classified based on the age estimated by the SVR, to get the feature information learned by the *support vectors*. Finally bins are created and averaged for each age, to get a table of AAM parameters, that can be used to morph a face to a desired age.

Young H. Kwon and Niels da Vitorie Lobo [21] developed an algorithm based on ratios of different facial features and a wrinkle analysis, including the automatic extraction of the required features. Their approach only needs a manually initialized center position of the head to fit an oval around the face. With this information initial position of iris, mouth and nose are set and optimized with the image potential technique. This information allows the computation of different ratios, which they discussed regarding to their reliability and robustness. For the wrinkle analysis they searched in different regions, like eyes and forehead, by dropping randomly oriented snakelets [20] to these regions. For the classification, the ratios are used to differentiate between baby and non baby and the wrinkles between adults and seniors. So for example a person, who is not a baby and has no wrinkles is an adult.

Sethuram et al. [42] researched the AAM performance related to facial aging. They compared the performance of building one general model and having a model for every ethnicity, gender and age group combination. For the evaluation they used the two ethnic groups American and African and the two age groups 18-45 and 46-65 years. Further the FaceVACS©SDK [7], a commercial face recognition system was used to generate a match score, which showed that the performance of the individual models was consistently better.

1.3. Aging Databases

There are a few publicly available databases which have facial images with age information. The three best known are used in many works and have already been mentioned in the related work Section 1.2. In the following we provide some basic description for each of them.

FG-NET: The database FGnet (Face and Gesture Recognition Research Network) was built by the European work group on face and gesture recognition. The database contains on average 12 pictures of varying ages between 0 and 69, for each of its 82 subjects. Altogether there are a mixture of 1002 color and greyscale images, which were taken in totally uncontrolled environments. Each was manually annotated with 68 landmark points. In addition there is a data file for every image, containing type, quality, size of the image and information about the subject such as age, gender, spectacles, hat, mustache, beard and pose. Some example images with landmark annotations are shown in Figure 1.3.

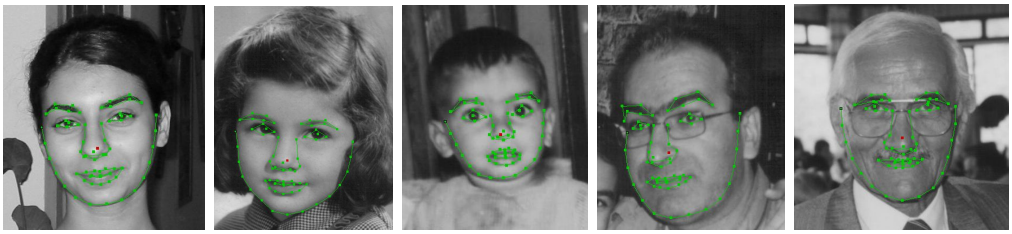


Figure 1.3.: Some images of the FG-NET database with landmarks

MORPH: The database was collected by the Face Aging Group and is intended for researchers interested in face-based biometrics. The database consists of two parts Album 1 and Album2. Album 1 contains 1690 greyscale images of 631 subjects between 15 and 68 years old. For every sample there is additional information about race, gender, facial hair, glasses, age and also 4 coordinates for the position of the eyes. Album 2 consists of 55608 images of 13673 subjects between 16 and 99 years. Information about race, gender, facial hair, glasses and age is available.

CVL: The Center for Vital Longevity (CVL) database [33] was created at the University of Michigan by Meredith Minear and Denise Park. The database can be divided into three categories. First, there is a set of 308 profile images. The next category consists of 580 neutral frontal face color images and an additional 159 greyscale pictures. The last one is divided into the emotions angry, annoyed, disgusted, grumpy, happy, sad, and surprised and contains a total of 258 images. The subjects belong to different races and are in the age range of 18 to 94. The gender and the three categories are given by the folder structure and the age is part of the filename.

1.4. Approach

An *Active Appearance Model* (AAM) is used to extract the appearance features, which is able to describe the shape and texture of an object with a set of parameters, once it has been trained for this type of object. So for this work the model is trained on a set of facial images. For learning the human aging process and the following age estimation, the popular machine learning method *Support Vector Machine* (SVM) is used. The estimation is done in two steps. At first a binary classification between youths and adults is done and in a second step the exact age is estimated by a more specific classifier based on the result of the first step.

Further some modifications are made to the age estimation system. Instead of the AAM's texture representation, *Discrete Cosine Transform* (DCT) is used to extract the texture appearance features. The first classification step between youths and adults is dropped, having only one global classifier for the whole age range. Instead of dropping the first classification step it is also tried to soften the classification by treating close decisions separately.

1.5. Outline

Starting in the Sections 2.1 and 2.2 the two basically used techniques *Active Appearance Models* (AAM) for feature extraction and *Support Vector Machines* (SVM) for machine learning are introduced. The intention is to provide a brief insight into the theoretical background, which is relevant for later application. Details about the feature extraction using AAM and the construction of the age estimation system including the implementation are provided in Section 3.

In Section 4.1 several tests were run and evaluated with different performance measures, commonly used in the context of age estimation. It appears that the classifier system can compete with other currently available age estimation techniques. Several modifications to further improve the prediction accuracy are introduced and evaluated in Section 4.2. A performance comparison to other approaches is done in Section 4.3. Finally the real life application of AAM and the consequences for the age estimation are discussed in Section 4.4.

2. Methodology

2.1. Active Appearance Models

Active Appearance Models (AAM) [46] rely on the so called *Active Shape Models* (ASM) [3], where the *Principal Component Analysis* (PCA) is used to get a statistical model of appearance for an object in a lower dimensional space. To create this statistical model an annotated training set of images is needed, where the points describe the shape of the object. After training, any shape s can be approximated by a set of parameters b_s using the formula $s = \overline{s} + P_s b_s$, where $\overline{s}$ is the calculated mean shape and P_s is the learned orthogonal modes of variation. The accuracy of this approximation depends on the number of kept eigenvectors, which is normally determined by the proportion of variance in the training set that should be described by the model. When fitting the shape, starting with the mean shape, the surface of each point is reviewed to find the best new position. One possible measurement for the quality of this position is the strongest nearby edge. Then the pose and shape parameters are updated by varying b_s in the learned ranges to get a new plausible shape approximation. This is repeated until convergence is reached, which is declared when an iteration step leads to no significant change of the parameters.

AAM now also include the texture of the object. The common problem that the pose of the object leads to texture changes, which dominate all other texture information, is solved by using the calculated mean shape and a triangulation algorithm, to warp the object into a shape free version (see Figure 2.1). After normalizing the texture in the shape area, PCA is used to get a statistical model of texture variation. So instead of the shape in ASM any texture t can be approximated by $t = \overline{t} + P_t b_t$, where $\overline{t}$ is the mean color level, P_t is the orthogonal modes of variation and b_t is the color level parameters. When synthesizing an object the parameters of b_t are first used to generate the shape free texture and in a second step b_s is used to warp the texture to match the shape.

The last step is to combine the shape and texture, which further reduces the dimension. For example in a facial image a big grin will not only variate the shape of the mouth, it will also cause the teeth to appear in the texture. To get a combined model the first step is to investigate the relationship between b_s and b_t. This is done with a weight matrix W_s, which is estimated during the training by displacing the shape from the optimum and observing the changes of the texture. So for every image a vector $b = \binom{W_s b_s}{b_t}$ can be created, which concatenates b_s and b_t. Once again PCA is used for the dimension reduction leading to a final model $b = Qb_c$, where Q is the matrix of eigenvectors and b_c is the combined vector of appearance parameters. Shape and texture can now be expressed by the formulas $s = \overline{s} + P_s W_s Q_s b_c$ and $t = \overline{t} + P_t W_t Q_t b_c$.

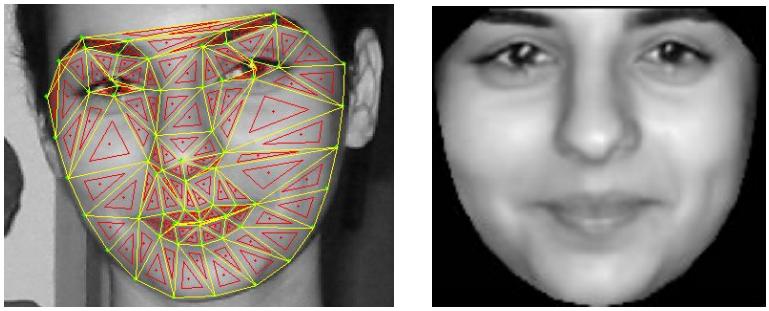


Figure 2.1.: Triangles used for warping (left) and shape free image (right)

2.2. Support Vector Machine

A *Support Vector Machine* (SVM) is a supervised learning method, which uses so called *support vectors* to build a model for classification or regression. The basic algorithm is described in V. Vapnik and A. Lerner's work [52]. The aim is to find an optimal hyperplane to separate two classes. In this case optimal means that besides just providing the lowest separation error, it is also as good as possible regarding generalization. This can be illustrated with a simple example, shown in Figure 2.2. The line H1 separates the two classes with no error, but the margin between the point clouds and the hyperplane is very small. In contrast H2 is also a separation with no error, but provides the greatest possible margin and thus promises the best generalization.

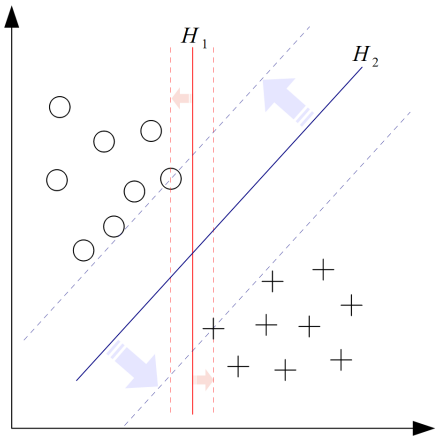


Figure 2.2.: Example for an optimal hyperplane

Let P be a linear separable two class problem and $d = \{(x_i, c_i)|x_i \in \mathbb{R}^m, c_i \in \{-1, +1\}\}$ one m dimensional vector of the training set D with $|D| = n$. To find the optimal hyperplane for P, an optimization problem has to be constructed. The basic concept is that two parallel hyperplanes h_1, h_2 are needed, which maximize the margin between the two classes. First of all any hyperplane can be written as a set of points x satisfying

the equation $\langle w, x\rangle + b = 0$, where w is the normal vector and $\langle \cdot, \cdot \rangle$ is the scalar product. To obtain a canonical form, the parameters w and b are scaled in a way that for each point $|\langle w, x\rangle + b| \geq 1$ is fulfilled. Thereby the points closest to the hyperplane are the ones, for which $|\langle w, x\rangle + b| = 1$ applies and hence describe the location of h_1 and h_2. Because of this, these points are the so called *support vectors* and give the procedure its name. The distance between h_1 and h_2 is $\frac{2}{\|w\|}$, which should be maximized and thus leads to the basic optimization problem to minimize $\|w\|$:

$$\min_{w,b} \|w\| \text{ subject to } c_i(\langle w, x_i\rangle + b) \geq 1, \forall i.$$

Because this optimization problem depends on $\|w\|$ it is difficult to solve. The substitution of $\|w\|$ with $\frac{1}{2}\|w\|^2$ leads to the following easier primal form:

$$\min_{w,b} \tfrac{1}{2}\|w\|^2 \text{ subject to } c_i(\langle w, x_i\rangle + b) \geq 1, \forall i.$$

The complexity of the constraints could also be reduced and leads to the following dual form:

$$\min_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2}\sum_{i,j} \alpha_i\alpha_j c_i c_j \langle w, x_i\rangle \text{ subject to } \sum_{i=1}^{n} \alpha_i c_i = 0,\ w = \sum_i \alpha_i c_i x_i,\ \forall \alpha_i \geq 0.$$

2.2.1. The Kernel Trick

The basic version of SVMs only allows linear classification, which can be changed by applying a so called kernel trick. The non-linear separable data is transformed into a higher dimensional space H using a mapping function ψ, in which the separation can be described linearly (Figure 2.3). Because H is often very high-dimensional, the calculation of the scalar products can be very complex. Now the kernel trick comes into play. It is based on the observation that ψ only appears in scalar products and can thus be replaced by a function k, which behaves like a scalar product and keeps the computational costs low. k is called the kernel function and it applies $k(x_i, x_j) = \langle \psi(x_i), \psi(x_j)\rangle$. Some commonly used kernel functions are:

- Polynomial: $k(x_i, x_j) = (x_i^T x_j + 1)^d$
- Radial: $k(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
- Gaussian Radial: $k(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$

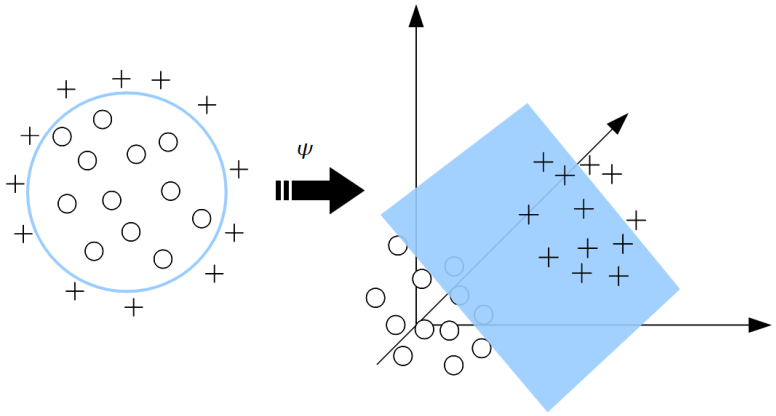


Figure 2.3.: Example for the transformation into a higher dimensional space. Separation in 2D space (left) is quite complex and can be described linearly after transformation into 3D space (right).

2.2.2. Soft Margin

The *Soft Margin* extension was introduced by C. Cortes and V. Vapnik [8]. It deals with the problem that outliers and random noise can easily lead the SVM to learn a hyperplane with a small margin between the classes and thus destroy the basic objective to build a general model. To prevent this, the basic optimization problem was extended as follows:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \text{ subject to } c_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \forall i, \xi_i \geq 0.$$

Where ξ_i measures the miss-classification of sample x_i and is called *slack variable*. So the new version searches for a hyperplane, which separates the examples as well as possible, but tolerates miss-classifications in order to maximize the margin between the classes. The weight between these two measures is controlled by the parameter C.

2.2.3. Regression (ϵ-SVR)

The *Support Vector Regression* (SVR) was introduced by V. Vapnik et al. [9] shortly after the soft margin extension. Its objective is also related, namely to find a flatter hyperplane. For this purpose, all training samples with an error less than ϵ are ignored. Hence deviations of samples near the hyperplane are not included into the loss function. So the training set is actually reduced, leading to a new basic optimization problem:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ subject to } \begin{cases} c_i - \langle w, x_i \rangle - b & \leq \epsilon \\ \langle w, x_i \rangle + b - c_i & \leq \epsilon \\ \forall i & \geq 0 \end{cases}$$

To combine the SVR with the soft margin extension, two *slack variables* ξ_i and ξ_i^* are introduced. The modified basic optimization problem is:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \text{ subject to } \begin{cases} c_i - \langle w, x_i\rangle - b & \leq \epsilon + \xi_i \\ \langle w, x_i\rangle - c_i & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \\ \forall i & \geq 0 \end{cases}$$

2.2.4. Multiclass Classification

The problem of building a SVM with more than two classes is normally solved by dividing it into multiple binary problems. There are two common methods for building such a classifier:

1. The *one-versus-all* method trains one classifier for each label l_i, which separates between l_i and all other classes. A new sample is assigned to the label with the highest classifier output.

2. The *one-versus-one* method trains one classifier for each pair of labels and the sample is assigned to the label with the most votes.

3. Implementation

3.1. AAM Building

For the AAM part of the work the *AAM-API* of Mikkel B. Stegmann [44] is used. The AAMs are trained on the FG-NET database, which provides a point-file with 68 landmarks for each of its 1002 images. In the first step it is necessary to convert these files, so that the *AAM-API* can handle them.

Configuration: The annotations shown in Figure 1.3 describe the shapes of the different face parts. In order to get the texture of the whole face, the AAM builder is configured to use the convex hull of these individual shapes (see Figure 2.1). The truncation level for the details represented in the shape, texture and the combined model is set to 95%. This means that the resulting models describe 95% of the variation appearing in the training set. For better generalization, especially for fitting the model without ground truth, it is important not to choose a too high truncation level. Finally the model size is halved leading to a mean texture area of about 13600 pixels, which still provides a detailed face representation and reduces the background noise of the pictures. For the whole configuration file, see appendix A

To get an impression what an AAM of the FG-NET looks like, the whole database was used to build the model. The combined feature vector b_c of the resulting AAM is only of size 48, which shows the power of the dimension reduction. The *AAM-API* orders the features by the percentage of variation they describe, of which the first 3 features of b_c already cover over 50%. A detailed overview, which also includes the shape and texture vector is listed in the model documentation file in appendix A.

For visualization the API allows the creation of synthesized images of the changes an individual feature causes. The result of doing this with the first five combined features in both directions of their learned standard deviation is displayed in Figure 3.1. It shows that the first four features mainly cover head pose and lighting variation, which explains their huge variation content. In contrast the fifth vector seems to contain a lot of age information, justifying the decision to use AAM features for age estimation.

3.2. AAM Fitting

Probably the most challenging part of AAM is fitting the model on an unseen image. For the FG-NET database the landmarks can be used to initialize the fitting. So in this case the position and size of the face is already known, which makes the fitting a lot easier. Because the focus of our work is the age estimation, the initialization of the AAM will be based on these landmark points, when the performance of the age classifier

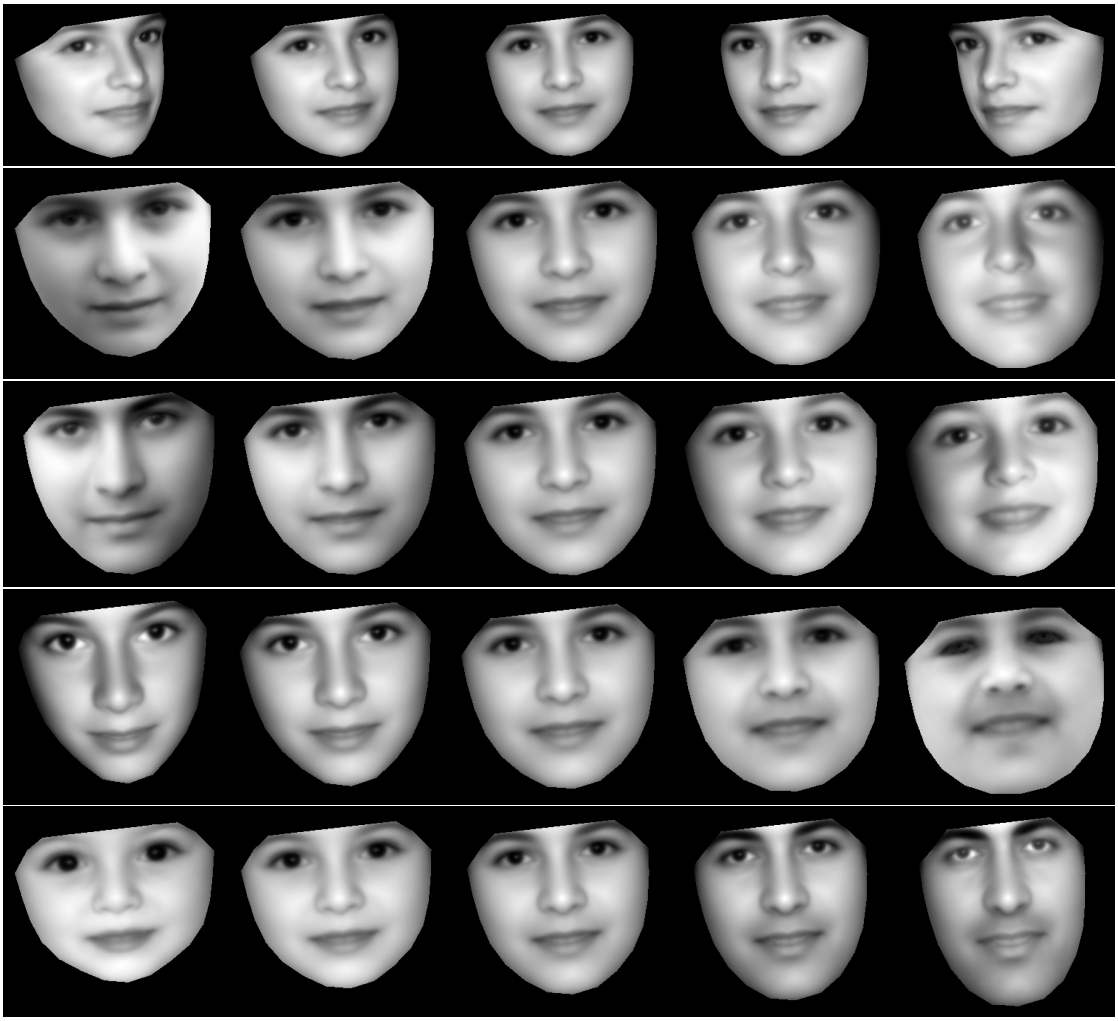


Figure 3.1.: Variation of the first five combined features in both directions, starting from the mean face, which is always displayed in the middle. AAM trained on the FG-NET database.

is measured. An approach for a full automatic system will be discussed in Section 4.4. Independent from the initialization method the following steps are performed for every iteration.

1. Calculate the current error vector $\delta g_i = g_s - g_m$, where g_s is the normalized image sample at the currently estimated shape and g_m the normalized grey-levels of the synthesized face.

2. Calculate the current error $E_i = |\delta g_i|^2$, using the *Mahalanobis distance*.

3. Compute the next displacement $\delta b_c = A\delta g_i$. A is the regression matrix, which is learned during the AAM training. It describes the parameter variations, which normally lead to convergence, which is important for an efficient iteration.

4. Calculate the new combined vector $b_{ci+1} = b_{ci} - k\delta b_c$, starting with $k = 1$.

5. Calculate the error vector δg_{i+1} for the new prediction b_{ci+1}.

6. If the new error $E_{i+1} < E_i$ accept the new estimation.

7. Otherwise go to step 4 and try a smaller step ($k = 1.5, 0.5, 0.25, ...$).

8. Repeat until the error is not further reduced or the maximum number of iterations is reached.

For visualization some pictures of the FG-NET database and their ground truth initialized fitting results are shown in Figure 3.2. The synthesized face is astonishingly close to the original, even recreating fine details like wrinkles, which is clearly visible in the last example of the second row.

Figure 3.2.: Some fitting results, AAM trained on the FG-NET database

3.3. Age Estimation

The basic construction of the age classifier is similar to the one in the work of Khoa Luu et al. [29]. The combined feature vector b_c of the AAM is used for a two step SVM classification. To use b_c in SVM it is important to first scale the parameters of the feature vectors to a fixed range, for example {-1,1}, to avoid that the parameters with huge ranges dominate the ones with smaller ones. The value with the biggest magnitude is determined for each parameter and is used as the scaling factor. In the first step, *Support Vector Classification* (SVC) is used for a binary classification between youths (under 21) and adults (21 and above), in future referred as the the youth/adult classifier. This border is chosen based on research on the anthropometry of the human face aging [2, 12, 32]. In summary in earlier years the growth of the face mainly causes changes in its shape. Approximately the age of 20 is a turning point from where mainly the texture changes as a consequence of wrinkles and sagging. In the second step, *Support Vector Regression* (SVR) is used for two separate classifiers, called the youth classifier and the adult classifier, to determine the specific age (see Figure 3.3). So all together three classifiers are trained, using the respective training data. For the youth/adult classifier all training images are used, whereas for the youth classifier only the youth faces and accordingly for the adult classifier only the adult faces are used.

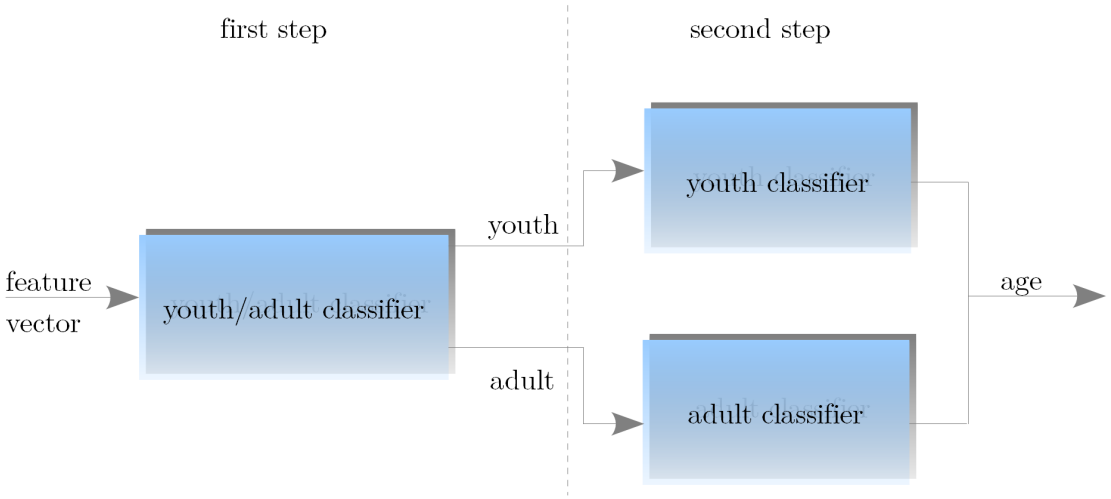


Figure 3.3.: The age classifier

For both, SVC and SVR the well known *LIBSVM* library [6] is used. All SVMs are configured to use a *Gaussian* kernel. Regarding the parameters, the stopping criteria ϵ is set to 0.001, whereas for SVC C and γ, and for SVR C, γ and p are optimized using a five fold cross validation.

In conclusion for building up the classifier the following procedure is processed:

1. Build an AAM on the training set.

2. Extract the feature vectors for all training images.

3. Scale the parameters of the feature vectors to {-1,1}.

4. Train the three classifiers using the respective data and optimize the SVM parameters on the training set.

And the estimation is done as follows:

1. Extract the feature vectors for all testing images.

2. Scale the parameters of the feature vectors with the same factors as the ones for training.

3. Finally estimate the age by applying the two step classification process.

4. Experiments

4.1. Evaluation

There are two common performance measures for age estimation, the *Mean Absolute Error* (MAE) and the *Cumulative Score* (SC). As the name implies, MAE is the mean difference of the predicted age and the real age and can therefore be calculated with the formula:

$$MAE = \frac{\sum_{i=0}^{n} |EA_i - RA_i|}{n}, \tag{4.1}$$

where EA_i is the estimated and RA_i is the real age for the i^{th} of n tested samples.

The *Cumulative Score* of an age difference d describes the percentage of estimations which have an estimation error of less than or equal to d years. This can also be described by the formula:

$$CS(d) = \frac{N(|EA_i - RA_i|)_{\leq d}}{n} \times 100, \tag{4.2}$$

where $N(|EA_i - RA_i|)_{\leq d}$ is the number of estimations with an estimation error less than or equal to d.

When optimizing the parameters with the cross validation the training and testing sets are chosen randomly in each fold. So the results can slightly variate between two optimization runs. Due to this fact all experiments were run several times, to get the average performance.

4.1.1. Comparison with Khoa Luu et al. work

Because the basic idea was from Khoa Luu et al. work [29], the first evaluation is a comparison to their results. For evaluation they used a static training set of 802 images from the FG-NET database, optimizing the SVM parameters via cross validation. The remaining 200 images were then used for testing the performance of the trained classifier. A diagram of the age distribution is presented in Figure 4.1. The imbalance in age ranges of the FG-NET database certainly also appears in the training and especially in the testing set, in which there are only 10 test samples over 40 years, 3 over 50 years and only 1 over 60 years of age. But more on this topic is at the end of this section.

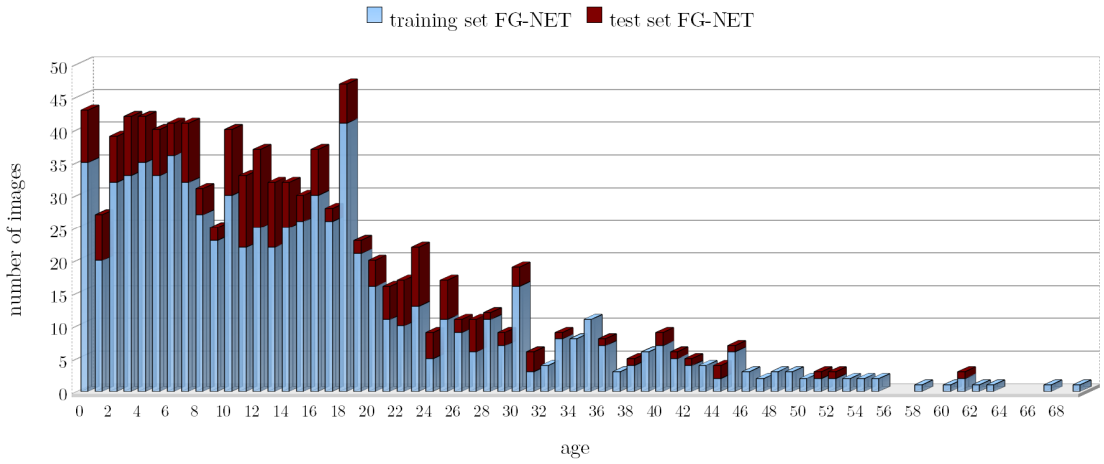


Figure 4.1.: Age distribution of the test and training set of Khoa Luu et al. work [29]

For evaluation the training and estimation procedure described in Section 3.3 were performed. The results are listed in Table 4.1. In the first test only the first 30 parameters of b_c were used, reaching an overall MAE of 4.25 years and in the second one all 46 parameters were used reaching an overall MAE of 3.98 years. So actually the features 31-46 contain age relevant information, even though they only cover a small part of the variation. In the columns 2-4 of Table 4.1 the performance of the single classifiers is considered separately.

- The youth/adult classifier has an error of 16% using the 30 dimensional feature vector and 17.5% using the 46 dimensional feature vector. This means that in the second step the wrong classifier is applied to 16% or 17.5% of the tested samples.

  To have an evaluation of the second step classifiers independent from the youth/adult classifier, for their calculation all samples are given to the correct second step classifier, ignoring the errors of the youth/adult classifier.

- The youth classifier reaches a very low MAE of 1.84 with the 30 dimensional feature vector and 1.69 years with the 46 dimensional one.

- The adult classifier is with an MAE of 5.83 years far above, but its age range is nearly twice as large and hence the expectancy value as well. In addition the adult classifier is trained only on 214 instead of 588 images. Because of the wider adult age range this should rather be the other way around. In the end the results are very close to the ones of Khoas Lue et al. [29], which indicates the functionality of our implementation.

When taking a closer look at the evaluation method, an unpleasant thing stands out. Besides the mentioned problem of having very few testing samples in the upper age range, there are images of the same persons in the training and testing set. So if there are two or more images of the same person with the same or nearly the same age, they are

features used	overall MAE (years)	y/a classifier error (%)	youth classifier MAE (years)	adult classifier MAE (years)
Khoa Luu et al.	4.37	-	1.93	5.80
combined 30	4.25	16.0	1.84	5.83
combined 46	3.98	17.5	1.69	5.83

Table 4.1.: MAE finding of our implementation on the training and testing set used in Khoa Luu et al. work [29]

likely to be similar, which can influence the results. The same problem also occurs, when randomly choosing the sets to optimize the SVM parameters, which may mislead the classifier to learn some "intra personal" relation. In consequence this evaluation method will no longer be used and has been replaced by a new one, introduced in Section 4.1.2.

4.1.2. Leave One Person Out

The *Leave One Person Out* evaluation method (LOPO) always selects all pictures of one person for testing, using all remaining samples as the training set. So in case of the FG-NET database this leads to 82 folds. For each of these folds the whole training and estimation procedure described in Section 3.3 is performed. When optimizing the SVM parameter via cross validation on the current training set, the subjects instead of the single images are now randomly chosen. Hence all pictures of one subject are either in the training or testing set, avoiding that the classifier learns some "intra personal" relations. In the end all 1002 estimations are summarized, assuring a much more stable performance evaluation, without any subject dependencies between the training and testing set.

MAE Findings:

For the first test the combined vector b_c with around 47 features (slightly variating from fold to fold) was used, leading to an overall MAE of 5.58 years. Compared to the results in Section 4.1.1 there is a significant overall MAE increase of over one and a half years, confirming the doubts about the first evaluation method. It seems that especially the adult classifier benefited from the dependencies between training and testing set, probably caused by the small number of subjects with adult images.

Aside from b_c, the AAM also provides separate shape and texture vectors, which were used for the following tests.

1. Only the shape vector b_s with around 27 features is used.
2. Only the texture vector b_t with around 102 features is used.
3. Both, b_s and b_t are used by concatenating them.

All results are listed in Table 4.2. As expected b_c and the concatenation of b_s and b_t have a better performance than only using b_s or b_t. The shape with its low dimensional

representation, at least reaches an overall MAE of 6.16 years. Further b_t outperforms b_s and is not far behind the combination of both. So the texture clearly seems to contain more age information than the shape. The concatenation of b_s and b_t leads to a slightly better estimation, which implies that shape and texture contain complementary information. Finally the combined feature vector b_c is also able to take advantage of both, reaching the best overall MAE of 5.58 years and has in addition a lower dimension than the concatenation of b_s and b_t.

A closer look at the second step classifiers shows that their results are very close to each other, so they are not the reason for the overall MAE changes. These are caused by the youth/adult classifier, where not only the error rate of 18.50% to 20.44% strongly influences the overall result, it also makes a difference if it miss-classifies a 69 or a 21 year old person. However the second step classifiers do not really seem to profit from the additional shape information, having nearly the same MAE as only using the texture vector. Neither does b_s improve the youth classifier nor does b_t achieve significantly better adult classification performance than b_s. What can be confirmed is that the combined texture and shape information allows a slightly better youth/adult classification, leading to a reduced overall MAE.

features used	overall MAE (years)	y/a classifier error (%)	youth classifier MAE (years)	adult classifier MAE (years)
shape only	6.16	20.44	2.32	7.77
texture only	5.84	19.24	2.15	7.55
shape & texture	5.71	18.84	2.16	7.55
combined	5.58	18.50	2.11	7.56

Table 4.2.: MAE for b_s, b_t, b_s concatenated with b_t and b_c, tested with the LOPO method on the FG-NET database

Age Range Analysis

To find the weaknesses of the classifier, the MAEs of the age ranges 0-9, 10-19,...,60-69 were calculated separately. All estimations are assigned to the age groups according to the real age. The results of this are listed in Table 4.3 and show that the increase of MAE in the upper age range is even more drastic than it seemed before. The reason for this is probably the same as for why the overall MAE is a lot lower - The number of images is drastically decreasing for the upper age range. So not only does the number of training samples get reduced, but the estimated MAE is also only based on a couple of images. The age range from 60-69 is the extreme case here with only 8 samples, which means that there is not even one image for each age. A look into the database shows that two of these images belong to person 3, two to person 4, one to person 5 and three to person 6. So in the 4 folds where the samples are tested the training set/testing set assignments are: 6/2, 6/2, 7/1 and 5/3, which makes the bad circumstances very clear.

Nonetheless the results contain a lot of information to analyze. To begin with, the

differences between the feature combinations besides from the shape only version are again very small. As discovered before, adding the shape information improves the youth/adult error. For the concatenation of b_t and b_c the resulting performance increase is as expected visible in all age ranges. In contrast for b_c the lower youth/adult error only improves the age ranges 0-9, 10-19 and 60-69, but in the others the performance is sometimes even worse than with b_t alone.

Because the differences between the feature combinations compared to the ones between the age ranges are marginal, a detailed analysis is only done for the combined feature vector. The first range, which includes people from the age 0 to 9 years has the lowest MAE of 2.28 years and is with 371 samples the largest group. Followed by the age range from 10-19 with 339 samples, which has an overall MAE of 5.01 years. This performance drop is not caused by the insufficient number of images, but by being close to the youth/adult border. In the range from 0-9 only 1.62% of the samples are given to the wrong second step classifier, but in the range from 10-19 there are 24.19%. The MAE increase of the range 20-29 with 144 samples compared to the range 10-19 is surprisingly small, considering that it is the first group that belongs to the adult classifier. The range 30-39 is expected to have a lower MAE than the range 20-29, because like the range 0-9 it is further away from the youth/adult border. But probably because there are only 79 images for training and testing, it is quite the opposite. The remaining ranges 40-49, 50-59 and 60-69 with 46, 15 and 8 samples confirm this assumption having a very huge performance drop.

features used	0-9 MAE	10-19 MAE	20-29 MAE	30-39 MAE	40-49 MAE	50-59 MAE	60-69 MAE
shape only	2.77	6.19	7.61	8.18	14.13	23.27	33.50
texture only	2.73	5.65	7.22	8.17	14.15	23.28	34.79
shape & texture	2.52	5.54	7.08	8.03	13.83	23.15	34.00
combined	2.28	5.01	7.29	8.31	14.11	23.54	33.3
image count	371	339	144	79	46	15	8

Table 4.3.: Age ranges MAE for b_s, b_t, b_s concatenated with b_t and b_c, tested with the LOPO method on the FG-NET database

Cumulative Scores:

The best representation for the *Cumulative Score* (CS) findings is a diagram, which contains the score for every examined age error. Figure 4.2 contains just this up to an age error of ten years. What is clear on first sight is that the CS confirms the MAE findings in terms of the performance order. But additionally the CS is able to show that over 30% of the estimations are already in an error range of only 1 year and over 70% in an error range of the overall MAE (5.5-6 years). Only b_s has a visible performance gap of approximately 5% to the other features.

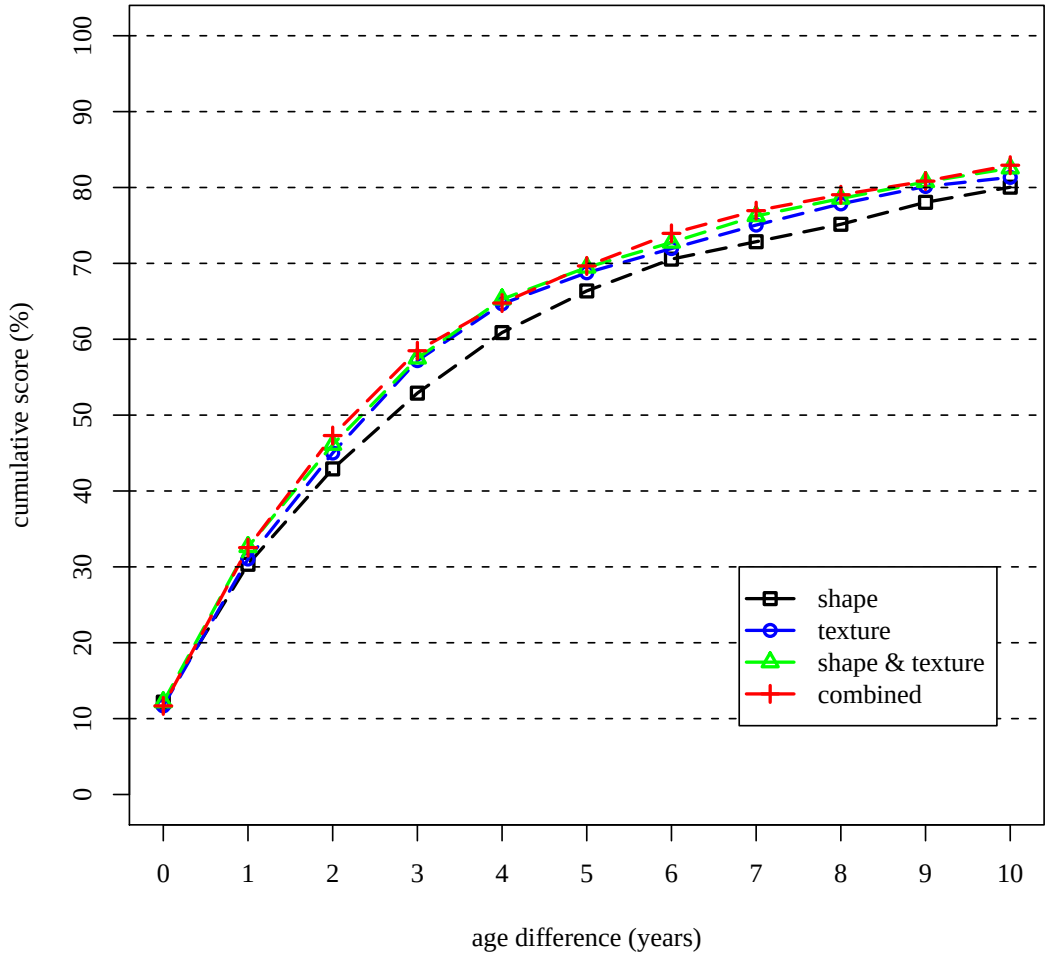


Figure 4.2.: Cumulative score comparison for b_s, b_t, b_s concatenated with b_t and b_c, tested with the LOPO method on the FG-NET database

Estimation Errors:

To get an impression of the single estimation errors, some of the most common are shown in Figures 4.3, 4.4 and 4.5. Besides the estimation errors caused by a younger or older appearance of the subject, bad image quality or uncommon head pose, there are also some which are not easy to explain. For these it generally appears that the estimation error is stable in a way that if a subject is estimated too young or old, this applies for more than one image.

Regarding the single classifiers the following things appear:

1. For the youth/adult classifier the most common errors are made around the age of 20, which is the critical region.

2. For the youth classifier no specific weaknesses could be found and large errors are rare and small compared to the adult classifier.

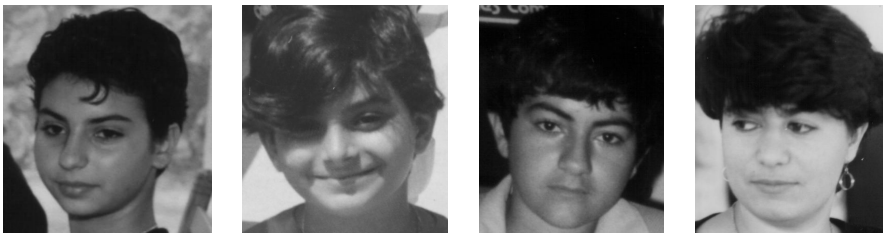
3. For the adult classifier all large estimation errors are made in the upper adult age range and due to the small number of training samples.

real age:	19	10	11	20
estimated age:	> 20	> 20	> 20	> 20

real age:	23	26	23	21
estimated age:	≤ 20	≤ 20	≤ 20	≤ 20

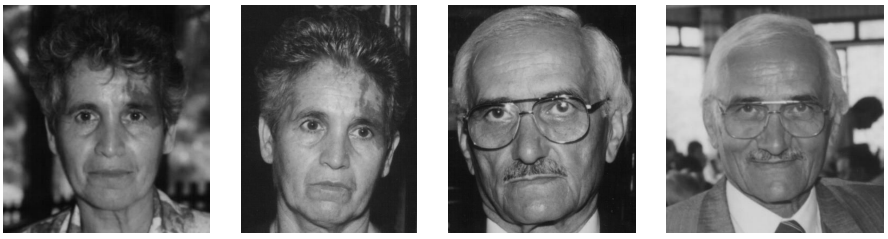
Figure 4.3.: Some youth/adult classification errors, tested with the LOPO method on the FG-NET database, using the combined feature vector

real age:	15	7	20	12
estimated age:	9	13	14	5



real age:	16	10	7	20
estimated age:	10	4	14	8

Figure 4.4.: Some large youth age estimation errors, tested with the LOPO method on the FG-NET database, using the combined feature vector



real age:	58	60	62	63
estimated age:	36	32	35	37

real age:	49	52	61	69
estimated age:	24	29	30	39

Figure 4.5.: Some large adult age estimation errors, tested with the LOPO method on the FG-NET database, using the combined feature vector

4.2. Modifications

4.2.1. Using DCT Features

One capability of an ASM is to produce a shape free version of a learned object, in this case the face. For AAM it is used to build the shape free texture model, so it should also be a good basis for other transformation methods like DCT. As a baseline the DCT feature extraction is done without the ASM. Instead, the eye coordinates are used to align the images, so that the face is cut out and the eyes are at a fixed position. A function for that is provided by the *OKAPI* library [22]. Some examples of the alignment are shown in Figure 4.6. The second version uses the shape free image for the extraction. Thus both versions only differ in this initial step. For the subsequent DCT extraction the following steps are performed using the DCT extractor of the *OKAPI* library [22]:

1. The generated image is scaled to 64×64 pixels.
2. The DCT is performed on blocks of 8×8 pixels.
3. For each block the first 5 coefficients in the *zig zag* scanning order are kept, leading to a $8 \times 8 \times 5 = 320$ dimensional feature vector.

   In contrast to AAM features, it is not necessary to scale the DCT parameters, because they are already normalized.

Figure 4.6.: Examples for the image alignment

Besides using only the feature vector b_{d_1} and b_{d_2} of the first and second version a combination of b_{d_2} and b_s is also used. The results of a LOPO evaluation with these feature combinations are listed in Table 4.4 together with the combined feature vector's MAE as a reference. The overall MAE of b_{d_1} is 0.26 years above the one of b_c. Apart from the higher dimensional feature vector, b_{d_1} only needs the eye coordinates for initialization and requires no fitting, so its performance is quite good for the much easier extraction procedure. The consideration of the single classifiers shows that compared to b_c the youth/adult error decreases, but the MAE of the youth classifier and adult classifier increases.

The overall MAE of b_{d_2} is equal to the that of b_c and this without using any shape information. Concatenating b_s decreases the overall MAE to a new optimum of 5.08 years, which is an improvement of 8.96% compared to the MAE of the combined feature vector. A look at the single classifiers shows that again the performance boost is caused by the youth/adult classifier. The error drops to 17.61% for b_{d_2} and to 15.65% for the

concatenation of b_{d_2} and b_s. Even the performance of the second step classifiers is slightly worse than using b_c, the overall MAE is better. The CS in Figure 4.7 confirm the MAE findings. Using b_{d_1} has the lowest CS for lower age errors, but the gap to b_{d_2} and b_c closes for age differences above 7 years. Nearly the same applies for b_{d_2}, which has a lower CS than b_c for age differences below 5 years. The concatenation of b_{d_2} and b_s has the highest CS and outperforms all other features.

features used	overall MAE (years)	y/a classifier error (%)	youth classifier MAE (years)	adult classifier MAE (years)
combined	5.58	18.50	2.11	7.56
DCT (Aligned)	5.91	17.42	2.75	7.85
DCT (ASM)	5.55	17.61	2.48	7.78
DCT (ASM) & shape	5.08	15.65	2.19	7.80

Table 4.4.: MAE for b_c, b_{d_1}, b_{d_2} and b_{d_2} concatenated with b_s, tested with the LOPO method on the FG-NET database

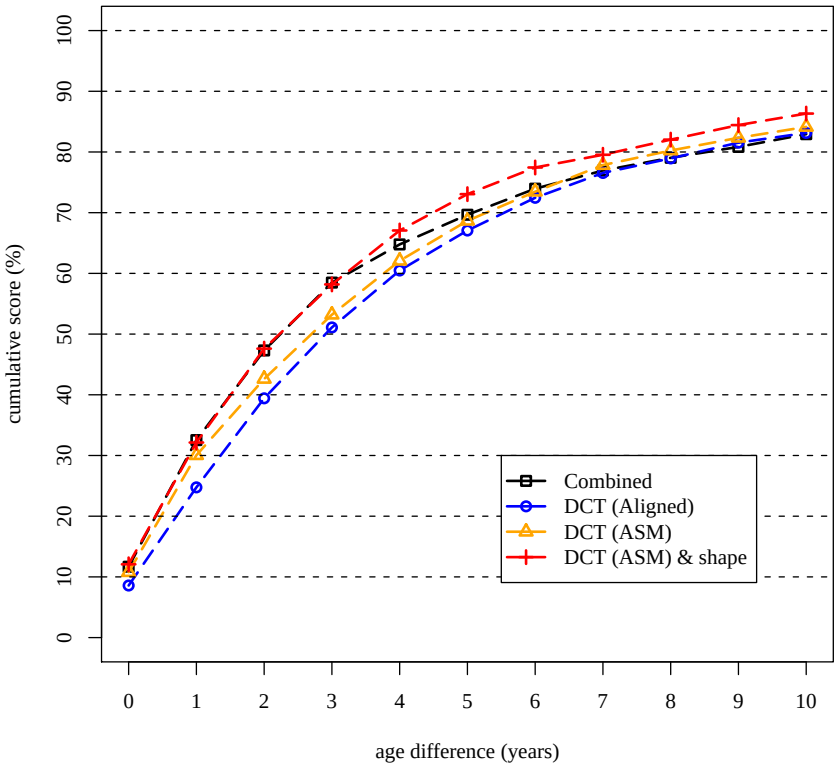


Figure 4.7.: Cumulative score comparison for b_c, b_{d_1}, b_{d_2} and b_{d_2} concatenated with b_s, tested with the LOPO method on the FG-NET database

4.2.2. Global Classifier Modification

During the evaluation the youth/adult classifier turned out to be a major weak point. Using the concatenation of b_{d_2} and b_s reduced the error to 15.56%, but this is still very high, considering that the same feature combination with a youth/adult error of 0% would reach an overall MAE of 3.74 years instead of 5.08 years. For b_c it would even decrease by 1.88 years to an overall MAE of 3.7 years. The idea is now to drop the youth/adult classification and just have one global classifier for the whole age range from 0 to 69 years.

The modification was tested with the LOPO evaluation using the best feature combinations of both previous parts, b_c and the concatenation of b_{d_2} and b_s. With 5.50 years b_c nearly reaches the same overall MAE as in the two step classification. This also applies for the concatenation of b_{d_2} and b_s with an overall MAE of 5.08 years, which questions the more complex two step classifier construction. A closer look at the results shows that the combination of having only one classifier and the unbalanced data set leads to the preference of youth classifications. The examination of the age ranges in Table 4.5 makes this clear. For b_c the ranges 10-19 and 20-29 benefit from dropping the youth/adult classification, because there is no longer a critical region. An interesting result of this is that the age range's MAE is now inversely proportional to the number of images in this range. As a result the MAE of the upper age ranges is significantly higher, so that the age estimation error is even more unbalanced compared to the "two step" classification. The same applies for the concatenation of b_{d_2} and b_s. Hence the "two step" classification is a better choice due to the more balanced age range errors.

features used	0-9 MAE	10-19 MAE	20-29 MAE	30-39 MAE	40-49 MAE	50-59 MAE	60-69 MAE
combined 2 step	2.28	5.01	7.29	8.31	14.11	23.54	33.3
combined 1 step	3.12	3.58	4.96	10.85	18.32	27.75	38.46
DCT (ASM) & shape 2 step	1.99	4.04	7.12	9.37	13.62	21.79	28.66
DCT (ASM) & shape 1 step	3.09	3.69	3.80	8.89	16.76	24.89	35.38
image count	371	339	144	79	46	15	8

Table 4.5.: Age ranges MAE for b_c and concatenation of b_{d_2} and b_s, tested with the LOPO method on the FG-NET database

4.2.3. Soft Youth/Adult Classification

The experiments in Section 4.2.2 showed that dropping the youth/adult classifier is no solution. The new approach is to modify the youth/adult classifier in a way to make the hard youth/adult decision softer. Therefore the decision value of the SVM is used, which is an indicator for the classification confidence and its value is near zero for samples which are near the decision border. So if the decision value is below a limit δ the decision is rated as close. In these cases the age estimation is then done by the global classifier,

which thus acts as a third second step classifier as illustrated in Figure 4.8. The value of δ is learned during the parameter optimization of the youth/adult classifier. All decision values of classified samples of the ages 20 and 21 are averaged, because they are expected to be close decisions.

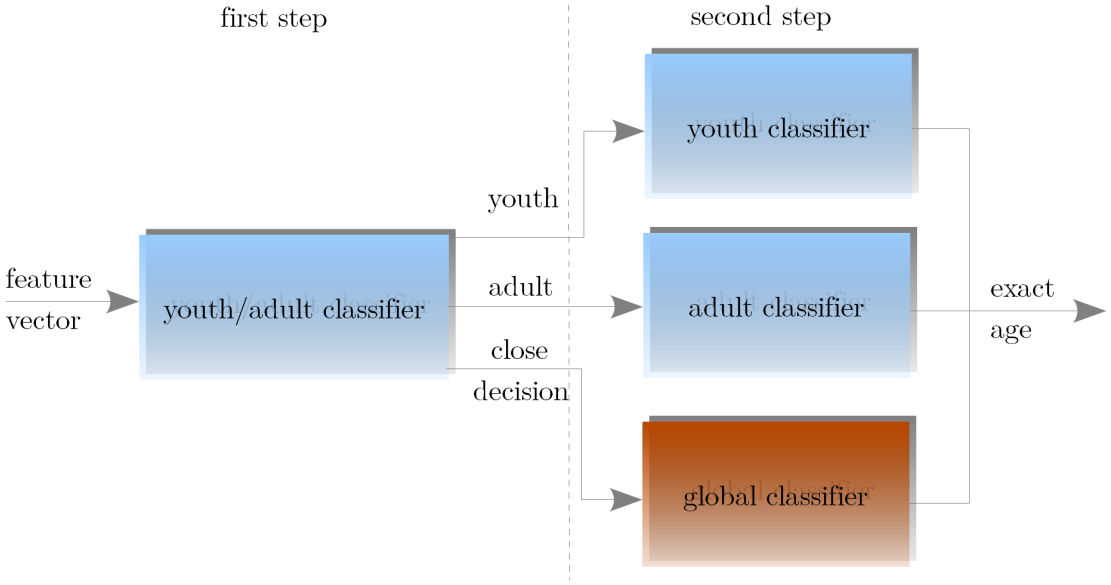


Figure 4.8.: The age classifier with the soft youth/adult classification

The modification was again tested with the LOPO evaluation using the feature combinations b_c and the concatenation of b_{d_2} and b_s. The combined feature vector reaches an overall MAE of 5.21 years. For the concatenation of b_{d_2} and b_s the modification leads to nearly the same improvement of 6.1%, resulting in a new optimum overall MAE of 4.77 years. If the samples which are given to the global classifier are not counted as a false classification, the youth/adult error decreases to 5.69% for b_c and 4.59% for the concatenation of b_{d_2} and b_s. Table 4.6 shows that the MAE of the age ranges 10-19 and 20-29 has improved as expected. Like in the *global classifier* modification there is a MAE decrease in other age ranges, but by far not as much.

features used	0-9 MAE	10-19 MAE	20-29 MAE	30-39 MAE	40-49 MAE	50-59 MAE	60-69 MAE
combined	2.41	3.88	5.78	9.92	15.83	25.64	35.46
DCT (ASM) & shape	2.19	3.67	4.97	9.10	15.09	22.60	32.13
image count	371	339	144	79	46	15	8

Table 4.6.: Age ranges MAE for b_c and concatenation of b_{d_2} and b_s with soft youth/adult classification, tested with the LOPO method on the FG-NET database

The CS in Figure 4.9 confirms this improvement. Compared to the basic version the

CS of the modification is always at least equal. To be more precise, for b_c the modification has a higher CS above an age difference of 4 years and for the concatenation of b_{d_2} and b_s the CS is higher above an age difference of 7 years.

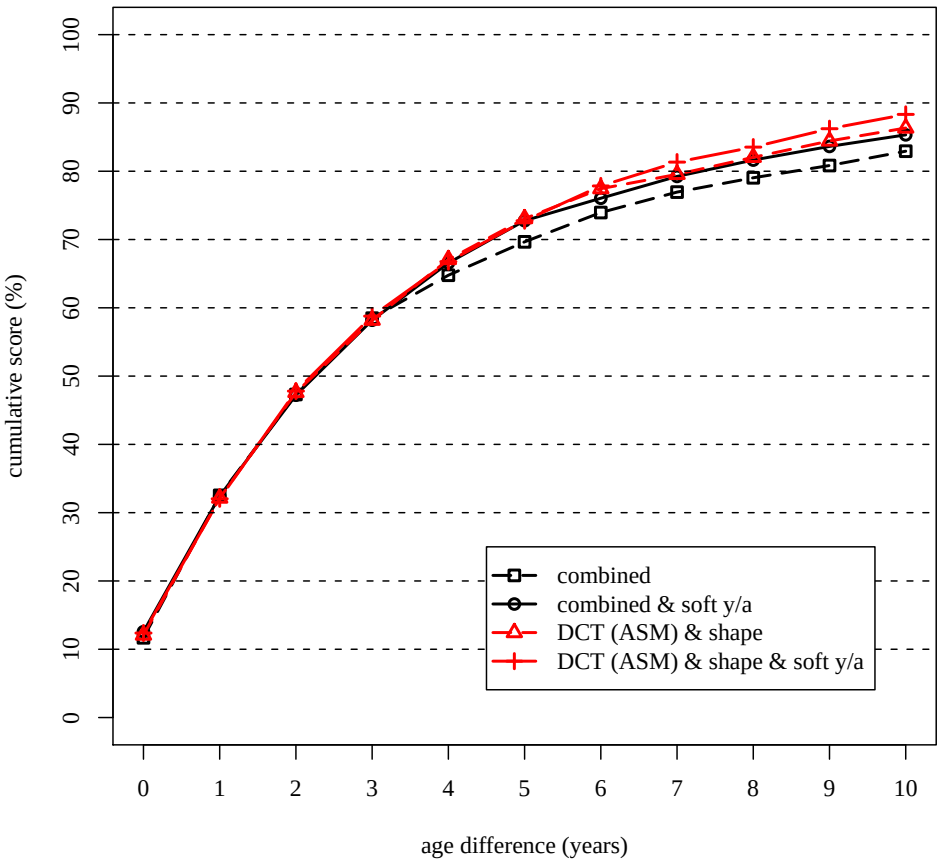


Figure 4.9.: Cumulative score comparison with and without soft youth/adult classification for b_c and b_{d_2} concatenated with b_s, tested with the LOPO method on the FG-NET database

4.3. Comparison to Other Estimation Techniques

Because the LOPO evaluation on the FG-NET database is used in other works, the results can be directly compared, which is done in Table 4.7. The compared methods are AGES$_{lda}$ [18], KAGES [16], WAS [24], AAS [23], kNN [31], mkNN [53], MSA [15], LARR [19], RUN1 [54], RUN2 [55], SVR [56], GP [56], WGP [56], and MTWGP [56]. It appears that the proposed classifier system with the concatenated DCT and shape vector and the soft youth/adult classification reaches the lowest MAE on the FG-NET database. The corresponding CS comparison in Figure 4.10 confirms this result.

Also very interesting is a comparison to the human skills of age estimation. For this, the research results of Xin Geng and Zhi-Hua Zhou work [18] are used. 51 random face images of the FG-NET database were given to 29 test candidates, leading to a total of $51 \times 29 = 1479$ estimations for the MAE calculation. In the first test only the greyscale image of the face area on which the AAM works was presented to the subjects, leading to a MAE of 8.06 years. In a second test the candidates could see the whole color image, resulting in a much better MAE of 6.23 years. Nevertheless our proposed classifier system is able to beat both results and is clearly better when additionally using the DCT and soft youth/adult classification.

Technique	MAE (years)
combined	5.58
combined & soft y/a	5.27
DCT (ASM) & shape	5.08
DCT (ASM) & shape & soft y/a	4.77
AGES_{lda} [18]	6.22
KAGES [16]	6.18
WAS [24]	8.06
AAS [23]	14.83
kNN [31]	8.24
mkNN [53]	4.93
MSA [15]	5.36
LARR [19]	5.07
RUN1 [54]	5.78
RUN2 [55]	5.33
SVR [56]	5.91
GP [56]	5.39
WGP [56]	4.95
MTWGP [56]	4.83
HumanA [18]	8.13
HumanB [18]	6.23

Table 4.7.: MAE comparison to other age estimation techniques, using LOPO method on the FG-NET database

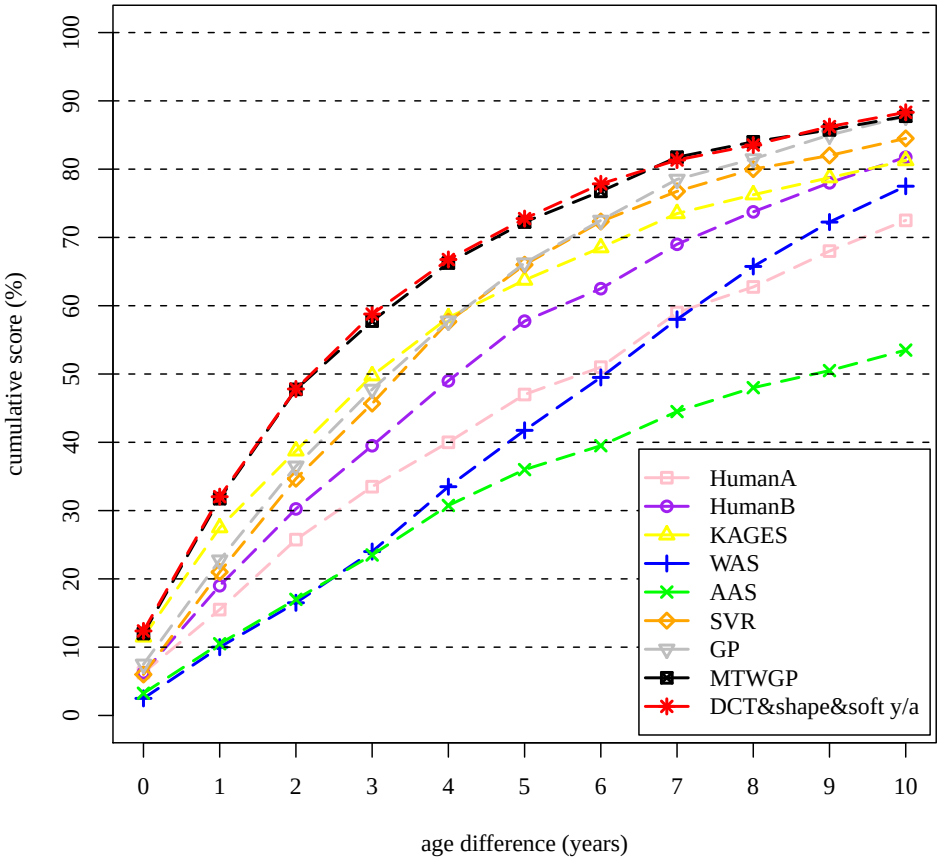


Figure 4.10.: Cumulative scores comparison to other age estimation techniques, tested with the LOPO method on the FG-NET database

4.4. Automatic Initialized Fitting

In real life there are no landmarks to initialize the AAM fitting, so there is no information about size or position of the face. Trying to fit the AAM at any possible position in any size will cost too much time and produce a lot of false positive detections. A better solution to deal with this is to use a face detector to get position and size of the face. In this work the MCT detector of the *OKAPI* library [22] is first used for face detection and then for eye detection. The output of the face detection rectangle is used to verify that the eyes are at the correct position of the face, which prevents false detections. After this the eye coordinates are used to calculate how the mean shape has to be transformed and stretched for the initialization. One example of this procedure is illustrated in Figure 4.11. Here only the fitting of the shape is displayed, but at the same time the texture as a part of the AAM is also fitted.

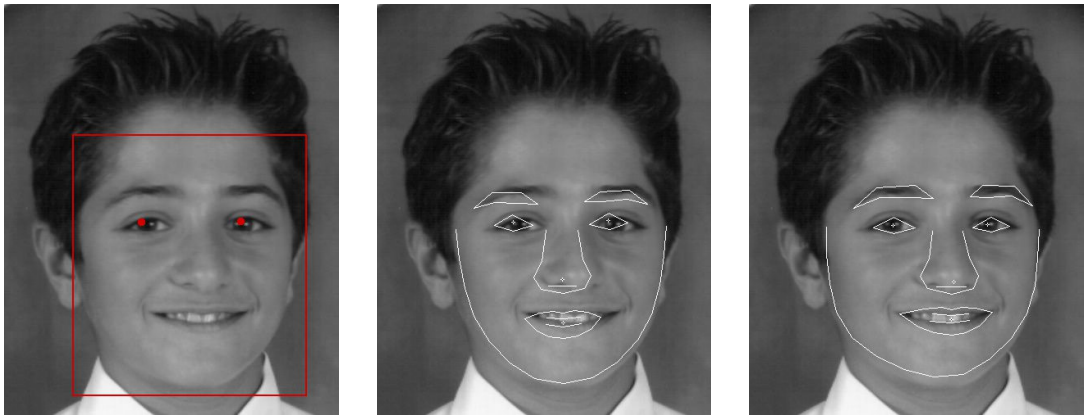


Figure 4.11.: Example for the automatic initialized AAM fitting procedure. 1. MCT face and eye detection (left), 2. initial shape (middle), 3. converged shape (right)

Using two detectors assures a high accuracy, but demands that both provide a correct result. Unfortunately this is not the case for 185 images of the FG-NET database, so they are excluded from testing, because no fall back strategies were implemented. The whole evaluation procedure stays nearly the same, with the small change that in the estimation procedure these new set of feature vectors is used.

4.4.1. Evaluation

To evaluate how much the automatic initialized fitting influences the estimation performance, all tests were run again with the new set of vectors. Because the automatic initialized fitting does not work for all images, the number of tested samples is reduced. To have comparable result all tests were also run on the reduced set using the ground truth initialized fitting. The results are listed in Table 4.8, showing that there is a MAE increase between 1.24 and 2.02 years. Especially the youth/adult and the youth classifier seem to be affected, while the adult classifier's MAE stays nearly the same.

Using only the shape has again clearly the weakest performance with an overall MAE of 7.83 years. The performance of b_t, b_t concatenated with b_s and b_c is with an overall MAE of about 7 years even closer together as before. The detection error of the eye coordinates has a strong influence on the estimation performance of the alignment based DCT extraction. The overall MAE increases by 1.95 years. The situation is completely different for the second version, which has with 1.24 years the lowest performance decrease. b_{d_2} alone reaches an overall MAE of 6.41, which is even further reduced to 6.12 years by concatenating it with b_s.

The soft youth/adult classification in Section 4.2.3 improves the overall MAE of b_c to 6.47 years, of b_{d_1} to 7.03 and of b_{d_2} concatenated with b_s to 5.67 years, which is the best result and has with 1.24 years the lowest decrease. The MAE of the youth and adult classifier of these tests are not listed, because of the mixture with the global

classifier. Apart from that it would be the same like in the basic version, since for the MAE calculation of the second step classifiers, all samples are always given to the correct classifier.

In Figure 4.12 the CS of the ground truth initialized and automatic initialized fitting is compared. This is done for b_c and the concatenation of b_{d_2} and b_s. For the investigated age differences the average CS decrease for b_c is 8.78% and for b_{d_2} and b_s concatenated it is 6.95%.

features used		overall MAE (y)	y/a class. error (%)	youth class. MAE (y)	adult class. MAE (y)
shape only	GI	5.81	20.38	2.25	7.33
shape only	AI	7.83	25.67	3.28	7.83
texture only	GI	5.31	17.63	2.10	7.05
texture only	AI	7.20	23.01	3.01	7.48
shape & texture	GI	5.17	17.01	2.08	7.03
shape & texture	AI	7.12	22.33	2.97	7.58
combined	GI	5.22	17.81	2.07	7.23
combined	AI	7.11	23.36	2.91	7.40
DCT (Aligned)	GI	5.32	16.28	2.56	7.35
DCT (Aligned)	AI	7.28	21.34	3.96	7.77
DCT (ASM)	GI	5.15	16.98	2.41	7.38
DCT (ASM)	AI	6.41	20.38	2.86	7.76
DCT (ASM) & shape	GI	4.74	16.01	2.14	7.34
DCT (ASM) & shape	AI	6.12	20.14	2.73	7.61
combined & soft y/a	GI	4.87	5.27	-	-
combined & soft y/a	AI	6.47	9.49	-	-
DCT (Aligned) & soft y/a	GI	5.07	6.44	-	-
DCT (Aligned) & soft y/a	AI	7.03	9.51	-	-
DCT (ASM) & shape & soft y/a	GI	4.43	5.08	-	-
DCT (ASM) & shape & soft y/a	AI	5.67	9.14	-	-

Table 4.8.: MAE comparison for all feature combinations with and without automatic initialized AAM fitting, tested with the LOPO method on the FG-NET database. GI stands for ground truth initialized and AI for automatic initialized AAM fitting.

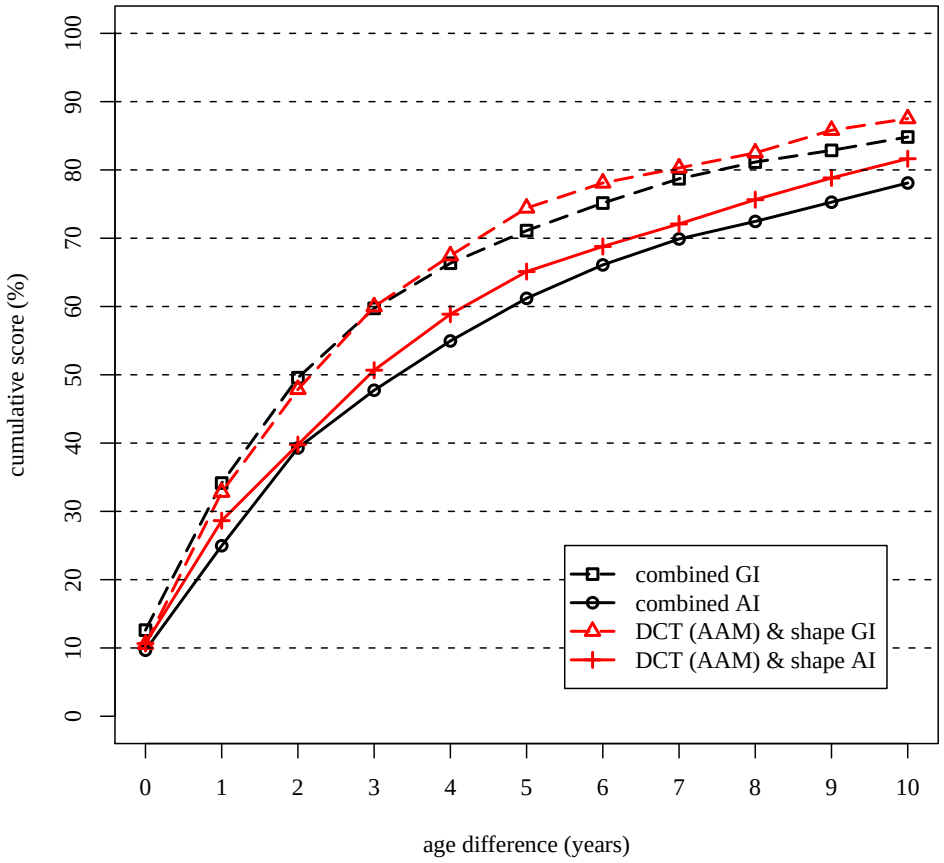


Figure 4.12.: Cumulative score of b_c and concatenation of b_{d_2} and b_s with and without automatic initialized fitting, tested with the LOPO method on the FG-NET database. GI stands for ground truth initialized and AI for automatic initialized AAM fitting.

5. Conclusion

This paper presented a facial image-based age estimation system that combines local appearance representation and shape information. A two step classifier is used for age estimation, where the first step performs a classification between youths and adults, and the second step performs the estimation of the exact age.

In Section 4.2.1 DCT was used as an alternative for the texture representation. Combined with the shape information it even turned out to perform better than the combined feature vector of the AAM, caused by the better youth/adult classification. Still the youth/adult error was with 15.65% very high, so in a second modification it was tried to drop the youth/adult classifier having only one global classifier for the whole age range. Surprisingly this had no impact on the overall performance, but was actually coping worse with the unbalanced age ranges. In a last modification an attempt was made to soften the youth/adult classification by using the global classifier for close youth/adult decisions, which improved the estimation performance. Combining the DCT extracted features and shape information and taking a soft decision for youth/adult classification attained a mean absolute error of 4.77 years, which is the lowest error reported on the FG-NET aging database when using the *Leave One Person Out* evaluation scheme.

The unbalanced number of images over the ages, which the FG-NET databases provided stood out as a huge problem for several times. Above the age of 60 years there is not even a sample for every age, which obviously makes training nearly impossible. The age range analysis in Sections 4.1.2 and 4.2.2 showed that the MAE is inversely proportional to the number of training images. So with an improved training set the proposed classifier system has the potential to provide a balanced and high-performance age estimation.

Finally the classifier's performance in a real life application was tested. This means that the AAM was fitted without using the landmark points from the database. The chosen initialization procedure, consists of a face and eye detection, was not successful for all images and no fall back strategies were implemented. So the performance comparison was done on a reduced testing set, on which an increase of the MAE by more than one year was observed.

5.1. Future Work

In a future work the proposed age estimation system performance should be tested on an extended data set. Images of one of the other aging databased listed in Section 1.3 could be used. Since manually annotating the images is very expensive, the AAM trained on the FG-NET database and the auto initialized fitting procedure introduced in Section 4.4 could be used to detect the landmark points. These then could be used

for a new AAM and age classifier training. Additionally the Mahalanobis distance as a quality measurement for the fitting offers the possibility to sort out bad fitting results and only take the accurate ones for the training. A further step would be to extend the data set with multi-ethnic face images.

Like shown in Sethuram et al. work [42] the AAM feature extraction can be improved by using specific AAM models for different age, gender, ethnic group, etc.. In case of this work's classifier system one possibility would be to use one global AAM for the youth/adult classification and two more specific models for the second step classifiers. Fitting the second model should also be simpler, because the same initialization shape or even the result shape of the global AAM fitting could be used.

One interesting option to make the SVM classification more accurate is feature selection. As shown in Figure 3.1 some age relevant features can be detected by simply looking at its effect on shape and texture.

Bibliography

[1] A. M. Albert and K. Ricanek. The morph database: Investigating the effects of adult craniofacial aging on automated face-recognition technology. *Forensic Science Communications*, 10(2), 2008. 1.2

[2] A. Midori Albert, Karl Ricanek Jr., and Eric Patterson. A review of the literature on the aging adult skull and face: Implications for forensic science research and applications. *Journal of Forensic Science International*, 172(1):1–9, 2007. 3.3

[3] M. Albert, K. Ricanek, and E. Patterson. Active shape models - their training and application. *Computer Vision Graphics and Image Understanding*, 61(1):38–59, 1995. 2.1

[4] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 711–720, 1997. 1.2

[5] M. Burt and D. I. Perrett. Perception of age in adult caucasian male faces: Computer graphic manipulation of shape and colour information. *Journal of Royal Society*, 259:137–143, 1995. 1.2

[6] Chih-Jen Lin Chih-Chung Chang. Libsvm: A library for support vector machines, 2001. 3.3

[7] Cognitec. Commercial face recognition system, 1995. 1.2

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1963. 2.2.2

[9] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161, 1997. 2.2.3

[10] G. J. Edwards, A. Lanitis, and C.J. Cootes. Statistical face models: Improving specificity. *Image and Vision Computing*, pages 203–211, 1998. 1.2

[11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004. 1.2

[12] L. G. Farkas. *Anthropometry of the Head and Face in Medicine*. Elsevier Science Ltd, 1981. 3.3

[13] FG-NET. Fg-net aging database. 1.2

[14] Feng Gao and Haizhou Ai. Face age classification on consumer images with gabor feature and fuzzy lda method. *Third International Conference on Advances in Biometrics*, pages 132–141, 2009. 1.2

[15] X. Geng and K. Smith-Miles. Facial age estimation by multilinear subspace analysis. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 865–868, 2009. 4.3, 4.7

[16] X. Geng, K. Smith-Miles, and Z.-H. Zhou. Facial age estimation by nonlinear aging pattern subspace. *16th ACM International Conference on Multimedia*, pages 721–724, 2008. 4.3, 4.7

[17] Xin Geng, Thi-Hua Thou, Yu Thang, Gang Li, and Honghui Dai. Learning from facial aging patterns for automatic age estimation. *In Proc. of 14th ACM Int'l Conf. Multimedia*, pages 307–316, 2006. 1.2, 1.2, 1.2

[18] Xin Geng, Zhi-Hua Zhou, and Kate Smith-Miles. Automatic age estimation based on facial aging patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2234–2240, 2007. 1.2, 4.3, 4.7

[19] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huan. Image-based human age estimation by manifold learning and locally adjusted robust regression. *IEEE Transactions on Image Processing*, pages 1178–1188, 2008. 4.3, 4.7

[20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988. 1.2

[21] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. *Computer Vision and Image Understanding Journal*, 74(1):1–21, 1999. 1.2, 1.2

[22] CVHCI lab. Okapi (open karlsruhe library for processing images) c++ library. 4.2.1, 4.4

[23] A. Lanitis, C. Draganova, and C. Christodoulou. Comparing different classifiers for automatic age estimation. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 34(1):621–628, 2004. 1.2, 4.3, 4.7

[24] A. Lanitis, C. J. Taylor, and T. F. Cootes. Modeling the process of ageing in face images. *Proceedings of the Seventh International Conference on Computer Vision*, 1:131–136, 1999. 4.3, 4.7

[25] A. Lanitis, C. J. Taylor, and T. F. Cootes. Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):442–455, 2002. 1.2

[26] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476, 2002. 1.2

[27] Khoa Luu, Tien Dai Bui, and Ching Y. Suen. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 1998. 1.2

[28] Khoa Luu, Tien Dai Bui, and Ching Y. Suen. Spectral regression based age determination. *IEEE Computer Society Workshop on Biometrics*, pages 103–107, 2010. 1.2

[29] Khoa Luu, Karl Ricanek, Tien D. Bui, and Ching Y. Suen. Age estimation using active appearance models and support vector machine regression. *Third IEEE International Conference on Biometrics*, pages 314–318, 2009. 1.1, 1.2, 3.3, 4.1.1, 4.1, 4.1.1, 4.1

[30] L. S. Mark, J. T. Todd, and R. E. Shaw. Perception of growth : A geometric analysis of how different styles of change are distinguished. *Journal of Experimental Psychology : Human Perception and Performance*, 7:855–868, 1981. 1.2

[31] E.A. Patrick and F.P. Fischer. A generalized k-nearest neighbor rule. *Information and Control*, 16(2):128–152, 1970. 4.3, 4.7

[32] E. Patterson, A. Sethuram, M. Albert, K. Ricanek, and M. King. Aspects of age variation in facial morphology affecting biometrics. *IEEE Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, 2007. 3.3

[33] Peter Peer, Borut Batagelj, and Jure Kovac andFranc Solina. Color-based face detection in the 15 seconds of fame art installation. *Conference on Computer Vision*, 2003. 1.3

[34] P. J. Phillips, H. Moon, and P. J. R. S. A. Rizvi. The feret evaluation methodology for face redcognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000. 1.2

[35] J. B. Pittenger and R. E. Shaw. Aging faces as viscal-elastic events: Implications for a theory of nonrigid shape perception. *Journal of Experimental Psychology : Human Perception and Performance*, 1(4):374–382, 1975. 1.2

[36] N. Ramanathan and R. Chellappa. Modeling age progression in young faces. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:387–394, 2006. 1.2

[37] N. Ramanathan and R. Chellappa. Modeling shape and textural variations in aging adult faces. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2008. 1.2

[38] Narayanan Ramanathan, Rama Chellappa, and Soma Biswas. Age progression in human faces : A survey. *Journal of Visual Languages and Computing*, 2009. 1.2

[39] K. Ricanek and T. Tesafaye. Morph : A longitudinal image database of normal adult age-progression. *IEEE International Conference on Automatic Face and Gesture*, pages 341–345, 2006. 1.2

[40] Karl Ricanek, Jr., Yishi Wang, Cuixian Chen, and Susan J. Simmons. Generalized multi-ethnic face age-estimation. *Third IEEE International Conference on Biometrics*, pages 127–132, 2009. 1.2

[41] Amrutha Sethuram, Eric Patterson, Karl Ricanek, and Allen Rawls. Improvements and performance evaluation concerning synthetic age progression and face recognition affected by adult aging. *IEEE Transactions on Image Processing*, 5558:62–71, 2009. 1.2

[42] Amrutha Sethuram, Karl Ricanek, and Eric Patterson. A hierarchical approach to facial aging. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 100–107, 2010. 1.2, 5.1

[43] R. E. Shaw, M. McIntyre, and W. Mace. The role of symmetry in event perception. *Perception: Essays in honor of James J. Gibson*, pages 276–310, 1974. 1.2

[44] B. Stegmann. The aam-api. 3.1

[45] J. Suo, F. Min, S. Zhu, S. Shan, and X. Chen. A multi-resolution dynamic model for face aging simulation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 1.2

[46] T.F.Cootes, G.J. Edwards, and C.J.Taylor. Active appearance models. *In Proc. European Conference on Computer Vision*, 2:484–498, 1998. 1.2, 2.1

[47] D. W. Thompson. *On Growth and Form*. Dover Publications, 1992. 1.2

[48] B. Tiddeman, M. Stirrat, and D. I. Perrett. Towards realism in facial image transformations : Results of a wavelet mrf method. *Computer Graphics Forum*, 24(3):449–456, 2005. 1.2

[49] B. Tidderman, D. M. Burt, and D. Perrett. Prototyping and transforming facial texture for perception research. *IEEE Computer Graphics and Applications*, 21(5):42–50, 2001. 1.2

[50] J. T. Todd, L. S. Mark, R. E. Shaw, and J. B. Pittenger. The perception of human growth. *Scientific American*, 242(2):132–144, 1980. 1.2, 1.1

[51] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 1995. 1.2

[52] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963. 2.2

[53] B. Xiao, X. Yang, Y. Xu, and H. Zha. Learning distance metric for regression by semidefinite programming with application to human age estimation. *17th ACM International Conference on Multimedia*, pages 451–460, 2009. 4.3, 4.7

[54] S. Yan, H. Wang, T. S. Huang, Q. Yang, and X. Tang. Ranking with uncertain labels. *IEEE International Conference on Multimedia and Expo*, pages 96–99, 2007. 4.3, 4.7

[55] S. Yan, H. Wang, X. Tang, and T. S. Huang. Learning autostructured regressor from uncertain nonnegative labels. *IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. 4.3, 4.7

[56] Y. Zhang and D.-Y. Yeung. Multi-taskwarped gaussian process for personalized age estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2622–2629, 2010. 4.3, 4.7

A. Appendix

AAM Configuration File

```
1  ##############################################################################
2  #
3  #   Active Appearance Model Builder Configuration File
4  #
5  ##############################################################################
6
7  2       # Model reduction               [1-n]   (reduction factor = 1/x)
8
9  0       # Model expansion               [0-n]   (pixels along the point normal)
10
11 1       # Use convex hull               [0|1]   (off/on)
12
13 0       # Verbose mode                  [0|1]   (off/on)
14
15 1       # Write registration movie      [0|1]   (off/on)
16
17 1       # Write variance image          [0|1]   (off/on)
18
19 0       # Produce model documentation   [0|1]   (off/on)
20
21 1       # Use tangent space projection  [0|1]   (off/on)
22
23 1       # Training method [ 0=PC Regression, 1=Jacobian (recommended) ]
24
25 95      # Shape model truncation (percentage [0-100], -1=parallel analysis)
26
27 95      # Texture model truncation (percentage [0-100], -1=parallel analysis)
28
29 95      # Combined model truncation (percentage [0-100], -2=no combined model)
30
31 1       # Subsampling of the training set (during training) [1-n]
32
33 1       # Warping method [ 0=benchmark, 1=software, 2=hardware (requires OpenGL) ]
```

AAM Documentation FG-NET Database

```
1  ######################################################################
2
3  Active Appearance Model File
4
5  Written                   : Saturday May 29 - 2010 [15:03]
6
```

```
Format version            : 0.99

Build time                : 12:23 (743.4 secs)

Shapes                    : 1002

Shape points              : 68

Texture Bands             : 1

Texture samples           : 13636

Texture TF                : identity

Model reduction           : 2

Add Shape Extents         : 0

Convex hull used          : Yes

Tangent space used        : Yes

Learning method           : 1

Shape truncation level    : 95 (variance: 0.0123/0.0129)

Texture truncation level  : 95 (variance: 1.53/1.61)

Combined truncation level : 95 (variance: 2.91/3.06)

Parameters used           : 48

Reference shape area      : 13652.35

Combined mode variation :
        1          30.17%         ( 30.17%)
        2          13.72%         ( 43.89%)
        3          11.25%         ( 55.14%)
        4           6.88%         ( 62.02%)
        5           5.51%         ( 67.54%)
        6           3.88%         ( 71.42%)
        7           2.93%         ( 74.34%)
        8           2.38%         ( 76.72%)
        9           1.58%         ( 78.30%)
        10          1.34%         ( 79.64%)
        11          1.16%         ( 80.79%)
        12          1.03%         ( 81.83%)
        13          0.96%         ( 82.78%)
        14          0.87%         ( 83.66%)
        15          0.75%         ( 84.40%)
        16          0.74%         ( 85.14%)
        17          0.62%         ( 85.77%)
        18          0.57%         ( 86.33%)
        19          0.54%         ( 86.87%)
```

```
          20            0.52%          ( 87.39%)
          21            0.48%          ( 87.87%)
          22            0.46%          ( 88.33%)
          23            0.45%          ( 88.78%)
          24            0.41%          ( 89.19%)
          25            0.40%          ( 89.59%)
          26            0.38%          ( 89.97%)
          27            0.37%          ( 90.34%)
          28            0.35%          ( 90.69%)
          29            0.33%          ( 91.02%)
          30            0.30%          ( 91.31%)
          31            0.29%          ( 91.60%)
          32            0.28%          ( 91.88%)
          33            0.26%          ( 92.14%)
          34            0.25%          ( 92.40%)
          35            0.24%          ( 92.64%)
          36            0.24%          ( 92.88%)
          37            0.23%          ( 93.11%)
          38            0.22%          ( 93.33%)
          39            0.21%          ( 93.54%)
          40            0.20%          ( 93.74%)
          41            0.19%          ( 93.93%)
          42            0.19%          ( 94.12%)
          43            0.18%          ( 94.30%)
          44            0.18%          ( 94.48%)
          45            0.18%          ( 94.66%)
          46            0.17%          ( 94.82%)
          47            0.16%          ( 94.98%)
          48            0.16%          ( 95.14%)

Shape mode variation :
          1            49.94%          ( 49.94%)
          2            12.45%          ( 62.40%)
          3             8.85%          ( 71.25%)
          4             4.37%          ( 75.62%)
          5             4.09%          ( 79.71%)
          6             2.12%          ( 81.83%)
          7             1.98%          ( 83.81%)
          8             1.72%          ( 85.53%)
          9             1.13%          ( 86.66%)
          10            0.94%          ( 87.61%)
          11            0.82%          ( 88.43%)
          12            0.72%          ( 89.15%)
          13            0.69%          ( 89.85%)
          14            0.65%          ( 90.50%)
          15            0.57%          ( 91.07%)
          16            0.50%          ( 91.57%)
          17            0.42%          ( 91.99%)
          18            0.42%          ( 92.41%)
          19            0.39%          ( 92.80%)
          20            0.36%          ( 93.16%)
          21            0.33%          ( 93.49%)
          22            0.32%          ( 93.81%)
          23            0.28%          ( 94.09%)
```

```
        24          0.28%        ( 94.37%)
        25          0.25%        ( 94.63%)
        26          0.24%        ( 94.87%)
        27          0.22%        ( 95.09%)

Texture mode variation :
        1          31.36%        ( 31.36%)
        2          21.06%        ( 52.42%)
        3           7.79%        ( 60.21%)
        4           3.34%        ( 63.54%)
        5           3.04%        ( 66.58%)
        6           2.66%        ( 69.24%)
        7           2.41%        ( 71.66%)
        8           1.49%        ( 73.15%)
        9           1.23%        ( 74.38%)
        10          1.18%        ( 75.56%)
        11          1.08%        ( 76.64%)
        12          0.86%        ( 77.50%)
        13          0.83%        ( 78.33%)
        14          0.78%        ( 79.11%)
        15          0.73%        ( 79.84%)
        16          0.69%        ( 80.53%)
        17          0.66%        ( 81.19%)
        18          0.62%        ( 81.81%)
        19          0.61%        ( 82.42%)
        20          0.51%        ( 82.93%)
        21          0.45%        ( 83.38%)
        22          0.45%        ( 83.83%)
        23          0.43%        ( 84.26%)
        24          0.43%        ( 84.69%)
        25          0.38%        ( 85.07%)
        26          0.36%        ( 85.43%)
        27          0.34%        ( 85.77%)
        28          0.33%        ( 86.10%)
        29          0.30%        ( 86.40%)
        30          0.29%        ( 86.70%)
        31          0.28%        ( 86.98%)
        32          0.27%        ( 87.25%)
        33          0.25%        ( 87.50%)
        34          0.25%        ( 87.75%)
        35          0.23%        ( 87.98%)
        36          0.22%        ( 88.20%)
        37          0.22%        ( 88.42%)
        38          0.20%        ( 88.62%)
        39          0.20%        ( 88.82%)
        40          0.19%        ( 89.01%)
        41          0.19%        ( 89.20%)
        42          0.18%        ( 89.37%)
        43          0.17%        ( 89.55%)
        44          0.17%        ( 89.71%)
        45          0.16%        ( 89.88%)
        46          0.15%        ( 90.03%)
        47          0.15%        ( 90.19%)
        48          0.15%        ( 90.33%)
```

49	0.15%	(90.48%)
50	0.14%	(90.62%)
51	0.14%	(90.76%)
52	0.13%	(90.89%)
53	0.13%	(91.02%)
54	0.13%	(91.15%)
55	0.13%	(91.27%)
56	0.12%	(91.40%)
57	0.12%	(91.51%)
58	0.12%	(91.63%)
59	0.11%	(91.74%)
60	0.11%	(91.86%)
61	0.11%	(91.97%)
62	0.11%	(92.08%)
63	0.11%	(92.18%)
64	0.10%	(92.28%)
65	0.10%	(92.38%)
66	0.10%	(92.48%)
67	0.10%	(92.58%)
68	0.09%	(92.67%)
69	0.09%	(92.77%)
70	0.09%	(92.86%)
71	0.09%	(92.95%)
72	0.09%	(93.03%)
73	0.08%	(93.12%)
74	0.08%	(93.20%)
75	0.08%	(93.28%)
76	0.08%	(93.36%)
77	0.08%	(93.44%)
78	0.08%	(93.51%)
79	0.08%	(93.59%)
80	0.07%	(93.66%)
81	0.07%	(93.74%)
82	0.07%	(93.81%)
83	0.07%	(93.88%)
84	0.07%	(93.94%)
85	0.07%	(94.01%)
86	0.07%	(94.07%)
87	0.06%	(94.14%)
88	0.06%	(94.20%)
89	0.06%	(94.26%)
90	0.06%	(94.32%)
91	0.06%	(94.38%)
92	0.06%	(94.44%)
93	0.06%	(94.50%)
94	0.06%	(94.55%)
95	0.06%	(94.61%)
96	0.05%	(94.66%)
97	0.05%	(94.72%)
98	0.05%	(94.77%)
99	0.05%	(94.82%)
100	0.05%	(94.87%)
101	0.05%	(94.92%)
102	0.05%	(94.97%)

```
223           103            0.05%          ( 95.02%)
224
225   ########################################################################
```