A Practical Subspace Approach To Landmarking

G. M. Beumer, and R.N.J. Veldhuis
Signals and systems group, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands
Email: {g.m.beumer,r.n.j.veldhuis}@ewi.utwente.nl

***Abstract*—A probabilistic, maximum aposteriori approach to finding landmarks in a face image is proposed, which provides a theoretical framework for template based landmarkers. One such landmarker, based on a likelihood ratio detector, is discussed in detail. Special attention is paid to training and implementation issues, in order to minimize storage and processing requirements. In particular a fast approximate singular value decomposition method is proposed to speed up the training process and implementation of the landmarker in the Fourier domain is presented that will speed up the search process. A subspace method for outlier correction and an iterative implementation of the landmarker are both shown to improve its accuracy. The impact of carefully training the many parameters of the method is illustrated. The method is extensively tested and compared with alternatives.**

***Index Terms*—Landmarking, eye, nose, mouth, localization, face recognition, outlier correction**

I. Introduction

A. Importance of registration for face recognition

Accurate registration is of crucial importance for good automatic face recognition. And although face recognition performance has improved greatly over the last decade [1], better registration will still lead to better recognition performance.

Many, but not all, registration systems use landmarks for the registration. A landmark can be any point in a face that can be found with sufficient accuracy and certainty, such as the location of an eye, nose and mouth. Some examples of landmarks are shown in Figure 1. The markers denote the landmarks as included in the BioID [2] database (left) or FRGC [3] database (right). Riopka et al. [4], Cristinacce and Cootes [5], Wang et al. [6], Campadelli et al. [7] and Beumer et al. [8], [9], and others have shown that precise landmarks are essential for a good face-recognition performance.

In [8], for example, it was shown that more accurate landmarking brings a higher recognition performance and that using more landmarks results in higher recognition performance. Besides face recognition there are other applications, such as positioning or measurement in an industrial setting, for which the detection of a landmark in an image with high accuracy is desirable.

B. Related work

Currently a popular approach is to use adaptations of the Viola-Jones [10] face finder for landmarking. We use a version of that method in this paper as a reference algorithm. The original Viola-Jones method uses weak Haar classifiers and a boosted training method known as Adaboost. Multiple variations to this have been proposed. For example, Wang et al. [6] use this method in combination with different classifiers for eye detection. Because the Haar classifiers only represent rectangular shapes they propose to use multiple weak Bayesian classifiers assuming Gaussian distributions.

Campadelli et al. [7] made a different variation on to the Viola-Jones classifier. They used a combination of Haar classifiers and Support Vector Machines to create an eye detector. The Haar classifiers do not work on the image texture but on their wavelet decomposition.

Cristinacce and Cootes [11] present a landmarking method called Shape Optimized Search where probability of the constellation of landmarks is used to predict where the landmarks are to be expected. Then, they use one of three different landmark detectors to refine the search.

Everingham and Zisserman [12] use three statistical landmarking methods, namely a regression method, a Bayesian approach and discriminative approach. The second method calculates a log likelihood ratio between landmark and background samples i.e. samples not containing a landmark. Everingham concludes that the Bayesian approach performs best compared with much more complicated algorithms. The Bayesian implementation is essentially the same as earlier work by Bazen et al. [13].

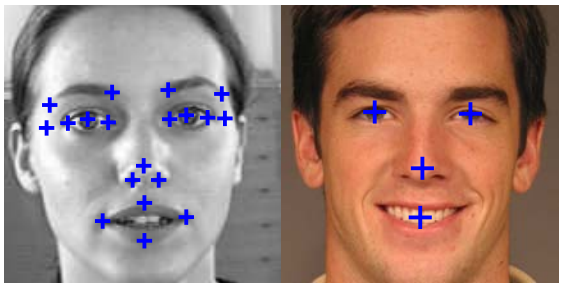

Fig. 1. Landmarks as provided by the BioID database (left) and the FRGC database (right).

C. Our work

In this paper we continue earlier work by Bazen et al. [13] and Beumer et al. [9]. A new theoretical foundation for the Most Likely Landmark Locator (MLLL) [9] is presented in Section II. This is followed by two practical

The work presented here was done in the contexts of the IOP-GenCom project BASIS and the Freeband-BSIK project PNP2008.

doi:10.4304/jmm.5.3.276-289

solutions for implementation problems that arise due to the size of the training data. First, an Approximate Recursive Singular Value Decomposition (ARSVD) algorithm is presented as a solution for computational limitations, regarding computer memory and processing time, which occur if the training data grows in volume. The ARSVD tackles this problem using subspaces. Second, a spectral implementation of MLLL will be derived, allowing for a more than tenfold speed-up of MLLL. These new modifications render MLLL a practical and accurate method for landmarking.

The application MLLL was designed for, is frontal face recognition with limited variation of pose and illumination. This implies that the landmarks will not be occluded, that they will be in predictable locations and that there will be no projective deformations. In more advanced versions of the proposed method, however, these constraints could be relaxed or dropped.

Two additions to MLLL are proposed. Namely, BILBO [9], which is a subspace-based outlier detection and correction algorithm for correcting erroneous landmarks. The first is a subspace-based outlier detection and correction method named BILBO that is capable of detecting and correcting erroneous landmarks. The second addition is a repetitive implementation of landmarking, The Repetition Of Landmark Locating (TROLL), which will improve accuracy. Both BILBO and TROLL can be used in combination with MLLL but can also work with any other landmarking algorithm. BILBO will be discussed in Section III and TROLL in Section IV.

MLLL, BILBO and TROLL all have parameters that have to be determined and that have a strong influence on the performance of the respective methods. In Section V we will analyse the relation of the parameters to the final performance of the algorithms.

An evaluation of the proposed methods and a comparison to other methods are presented in Section VI, showing that MLLL, especially with the extensions BILBO and TROLL, has a good performance. TROLL yields an error of 3.3% of the interocular distance. This error is obtained with a landmarker of which some of the parameters have not been optimized for specific landmarks, but for the entire set of landmarks. Tuning MLLL for each landmark individually is likely to improve the recognition performance further.

II. Most Likely Landmark Locator

In this section we will present the Most Likely Landmarks Locator. First, a theoretical framework for landmarking will be presented. After that some implementation issues will be addressed. In order to speed up the computations we introduce a frequency domain implementation. Also the Approximate Recursive Singular Value Decomposition (ARSVD) is presented as a solution for computing large volume databases using subspaces.

A. Theory

Let the shape $\vec{s}$ of a face be defined as the collection of landmark coordinates, arranged into a column vector. The texture samples of the face are within a region of interest (ROI) and also arranged into a column vector, $\vec{x}$. The maximum a posteriori estimate (MAP) [14] of the location of the landmarks, $\vec{s^*}$, given a certain texture $\vec{x}$, can be written as

$$\vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}}\, q(\vec{s}|\vec{x}), \tag{1}$$

where $q(\vec{s}|\vec{x})$ denotes the probability density of the shape given image $\vec{x}$. According to Bayes rule, Equation 1 can be rewritten as

$$\vec{s^*} = \underset{\vec{s}}{\operatorname{argmax}}\, \frac{p(\vec{x}|\vec{s})}{p(\vec{x})} q(\vec{s}), \tag{2}$$

where $p(\vec{x}|\vec{s})$ can be recognized as the probability density of the texture $\vec{x}$ given a shape $\vec{s}$. Furthermore, $p(\vec{x})$ denotes the probability density if the landmark locations are unknown. Finally, $q(\vec{s})$ is the probability density of the shape. The quotient in Equation 2 is the likelihood-ratio of the texture belonging to shape $\vec{s}$.

Ideally, one would like to compute $\vec{s}^*$ from Equation 2, including the prior probability density $q(\vec{s})$ of $\vec{s}$. In order to reduce the computational complexity we assume $q(\vec{s})$ to be uniform over the region of interest. Therefore $q(\vec{s})$ can be removed from Equation 2. Let $\vec{x}_i$ be the texture surrounding the i-th landmark and $\vec{s}_i$ its location. We assume, for practical reasons, that $\vec{x}_i$ only depends on $\vec{s}_i$ and that $\vec{x}_i$ and $\vec{x}_j$, $i \neq j$, are independent. Therefore,

$$\frac{p(\vec{x}|\vec{s})}{p(\vec{x})} = \prod_{i=1}^{l} \frac{p(\vec{x_i}|\vec{s_i})}{p(\vec{x_i})}. \tag{3}$$

With this simplification the optimization problem in Equation 2 can be reformulated as

$$\begin{aligned} \vec{s_i^*} \quad = \quad & \underset{\vec{s}}{\operatorname{argmax}} \sum_{i=1}^{l} \big(\log(p(\vec{x}_i|\vec{s}_i)) - \\ & \log(p(\vec{x}_i)) \big) \end{aligned} \tag{4}$$

We assume that the probability density of the landmark texture $p(\vec{x}_i|\vec{s}_i)$ is Gaussian with mean $\vec{\mu}_{l,i}$ and covariance matrix $\Sigma_{l,i}$. Likewise $p(\vec{x}_i)$, which we will denote as the background density, thus emphasizing that x_i may come from an arbitrary location, is Gaussian with mean $\vec{\mu}_{b,i}$ and covariance $\Sigma_{b,i}$. These assumptions have been made for practical reasons, but are mildly supported by the fact that especially after dimensionality reduction, the texture probability density tends towards Gaussian. A more accurate model might be a Gaussian mixture model, but that would be much more complex. Because of the assumed mutual independence of the landmarks, the terms in Equation 4 can be maximized independently. This makes that the estimation of the shape is now simplified to

$$\begin{aligned} \vec{s^*}_i \quad = \quad & \underset{\vec{s}}{\operatorname{argmax}} \Big\{ \big(\vec{x}_i(\vec{s}) - \vec{\mu}_{b,i}\big)^T \Sigma_{b,i}^{-1} \big(\vec{x}_i(\vec{s}) - \vec{\mu}_{b,i}\big) \\ & - \big(\vec{x}_i(\vec{s}) - \vec{\mu}_{l,i}\big)^T \Sigma_{l,i}^{-1} \big(\vec{x}_i(\vec{s}) - \vec{\mu}_{l,i}\big) \Big\} \end{aligned} \tag{5}$$

for all landmarks $i = 1 \ldots d$. This is identical to the op-

timization criterion used in MLLL presented in previous work [9]. Equation 5 is intuitively pleasing as each term of the summation benefits the similarity to a landmark and penalizes the similarity to the background.

1) Dimensionality reduction: The covariance matrices, Σ_l and Σ_b in Equation 5, need to be estimated from training data. Because landmark templates can be as large as $96 \times 64 = 6144$ pixels, direct evaluation of Equation 5 would be a too high a computational burden. Due to the limited number of training samples available in practice, the estimates of the covariance matrices could be rank-deficient. Even if not, they would be too inaccurate to obtain a reliable inverse, which is needed in Equation 5.

Therefore, prior to the evaluation of Equation 5, the vector $\vec{x}$ will be projected onto a lower dimensional subspace. This subspace should have several properties. First of all, its basis should contain the significant modes of variation of the landmark data. Secondly, it should contain the significant modes of variation of the background data. Finally, it should contain the difference vector between the landmark and the background means, for a good discrimination between landmark and background data. The modes of variation are found by principal component analysis (PCA) on landmark and background training data. After this first dimensionality reduction the landmark and background densities are simultaneously whitened [15], such that the landmark covariance matrix becomes a diagonal and the background covariance matrix an identity matrix in the reduced feature space. The latter whitening step is done for computational reasons. See Appendix A1 for details of the procedure of the dimensionality reduction.

The previous feature dimensionality reduction steps aimed at creating a good representation of the landmark and background data. In the next feature reduction step we want to select the features that have the highest discriminative power. In this feature selection step, a fixed number of features are kept. The standard Linear Discriminant Approach as proposed by Fisher [16] is not applicable because the covariance matrices $\Sigma_{b,i}$ and $\Sigma_{l,i}$ are different. Instead, our approach is to keep those features for which the mean divided by their standard deviation is maximal. Informal experiments in which this method was compared with alternatives have shown that this method gave the best results.

2) Feature extraction and classification: The total process of feature reduction and simultaneous whitening can be combined into one linear transformation by a matrix $T \in \mathbb{R}^{m \times n}$, with n the dimensionality of the training samples and m the final number of features after reduction. The detailed calculation of the feature reduction transformation T is given in Appendix A with the final result in Equation 32.

With T we project the means, covariance matrices and feature vectors onto the subspace, ideally:

$$\vec{\mu}_l' \stackrel{\text{def}}{=} T\vec{\mu}_l, \qquad \vec{\mu}_b' \stackrel{\text{def}}{=} T\vec{\mu}_b. \tag{6}$$

$$\Lambda_l \stackrel{\text{def}}{=} T\Sigma_l T^T, \qquad I_b \stackrel{\text{def}}{=} T\Sigma_b T^T. \tag{7}$$

$$\vec{y}(\vec{s}) \stackrel{\text{def}}{=} T\vec{x}(\vec{s}). \tag{8}$$

where Λ_l is diagonal, I_b is identity, $\vec{y}(\vec{s})$ is the feature vector and $\vec{x}(\vec{s})$ denotes sample values from the ROI at location $\vec{s}$. Please note that Σ and T are estimates obtained from data and, therefore, not exact. Consequently, the resulting covariance matrices after the transformation are only approximately diagonal. After this transformation Equation 5 becomes

$$\begin{aligned} \vec{s^*} \quad = \quad & \underset{\vec{s}}{\operatorname{argmax}} \Big\{ \big(\vec{y}(\vec{s}) - \vec{\mu}_b'\big)^T \big(\vec{y}(\vec{s}) - \vec{\mu}_b'\big) \\ & - \big(\vec{y}(\vec{s}) - \vec{\mu}_l'\big)^T \Lambda_l^{-1} \big(\vec{y}(\vec{s}) - \vec{\mu}_l'\big) \Big\}. \end{aligned} \tag{9}$$

Note that although Equation 9 resembles Equation 5, the result will be different due to the dimensionality reduction. Solving Equation 9 is, however, computationally far more efficient than solving Equation 5.

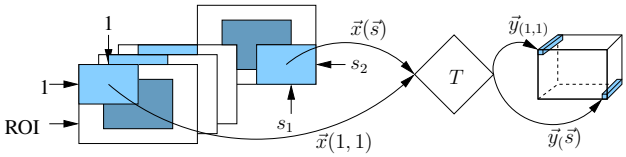

Fig. 2. Feature extraction in the spatial domain. The pixel values surrounding the location of interest, $\vec{x}(\vec{s}) \in \mathbb{R}^m$ are multiplied with $T \in \mathbb{R}^{m \times n}$. The resulting feature vector $\vec{y}(\vec{s}) \in \mathbb{R}^n$ is of lower dimensionality than $\vec{x}(\vec{s})$.

B. Approximate Recursive Singular Value Decomposition

Training on large data sets should make MLLL accurate and robust. However, as the amount of training data grows, the calculation of T quickly becomes computationally prohibitive, either because of time or, more likely, memory constraints. Especially the Singular Value Decompositions (SVDs) in Equations 21 and 29 in Appendix A1 are troublesome. In order to overcome these problems an Approximate Recursive SVD (ARSVD) algorithm is introduced. Proper application relies on two conditions. The first is that the estimates of the covariance matrix improve when more data is processed. Second, the amount of explained variance kept in each recursion step must be sufficient. As the SVD is part of the feature reduction process, finally only a certain amount of the explained variance is to be kept and the amount of variance kept by the ARSVD should be higher than that. If these two conditions are met, there should be no significant loss of information. ARSVD is fairly straightforward. Let X be a matrix with all feature vectors as columns, split up into a number of submatrices, called blocks, with a fixed number of columns, called the blocksize b:

$$X = [X_1, X_2 \ldots X_o] \tag{10}$$

Let U_j, W_j and V_j represent the ARSVD after j blocks, i.e.

$$[X_1 \ldots X_j] \approx U_j W_j V_j^T \tag{11}$$

with $U_j \in \mathbb{R}^{n\times n}$, $W_j \in \mathbb{R}^{n\times b}$ and $V_j \in \mathbb{R}^{b\times b}$. Note that the number of pixels in the samples, n is larger than the blocksize b. The space of $[X_1 \dots X_j]$ is spanned by $U_j W_j$. Adding the next block of data of X and calculating the SVD gives

$$\begin{aligned}[U_j W_j | X_{j+1}] &= \tilde{U}_{j+1}\tilde{W}_{j+1}\tilde{V}_{j+1}^T \\ &\approx U_{j+1}W_{j+1}V_{j+1}^T \end{aligned} \quad (12)$$

where $U_{j+1} \in \mathbb{R}^{n\times b}$ and $W_{j+1} \in \mathbb{R}^{b\times b}$ are submatrices of $\tilde{U}_{j+1} \in \mathbb{R}^{n\times n}$ and $\tilde{W}_{j+1} \in \mathbb{R}^{n\times 2b}$ of reduced sizes. Each run the dimensionality retained is reduced from twice the blocksize to the blocksize. Repeating this until all sub matrices of X are processed will give an estimate of the matrix U and matrix W after a standard SVD. The blocksize is a parameter that has an impact on the accuracy and the speed of the ARSVD.

C. Frequency domain implementation

Even in the reduced feature space, evaluating Equation 9 is still computationally demanding. This is because Equation 8 is evaluated at each possible location within a region of interest. A schematic overview of how the spatial algorithm operates is given in Figure 2. It can be observed that the calculation of each element of $y(\vec{s})$ is analogous to a filter operation or equivalently a cross-correlation operation. Hence we can make use of the fact that a cross-correlation operation in the spatial domain can be written as, a much less demanding, element wise multiplication in the spectral domain. The conversion to the spectral domain and back again can be done by an efficient implementation of a discrete Fourier transform, thus resulting in a net gain in processing time. As a result the processing time of an implementation in Matlab on a desktop PC was reduced more than tenfold.

Only considering the k-th element of vector $\vec{y}(\vec{s})$ from Equation 8 we have

$$y_k(\vec{s}) = \vec{t}_k \vec{x}(\vec{s}) \quad (13)$$

with $\vec{t}_k \in \mathbb{R}^{1\times n}$ the k-th row of $T \in \mathbb{R}^{m\times n}$. If $\vec{t}_k$ is reshaped to $\hat{t}_k \in \mathbb{R}^{v\times u}$ it can be seen as a correlation kernel, as seen in Figure 3, which is shifted over the ROI.

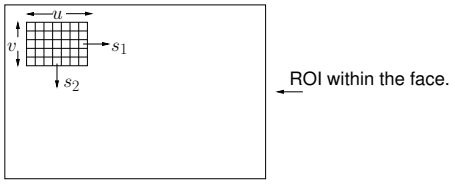

Fig. 3. Applying the kernel $\hat{t}_k$ to the image. The similarity between the kernel and the image is calculated at all locations (s_1, s_2). Each row in T can be considered to be a single kernel.

At each location $\vec{s}$ this can thus be written as:

$$y_k(\vec{s}) = \sum_u \sum_v \hat{t}_k(u,v) x(s_1+u, s_2+v). \quad (14)$$

Because correlation in the spatial domain corresponds to an element wise multiplication of the signal with the complex conjugate of the correlation kernel in the spectral domain [17], we get:

$$\begin{aligned}\mathcal{F}(y_k(\vec{s})) &= \mathcal{F}(\hat{t}_k(\vec{s}))\mathcal{F}(x(\vec{s}))^* && (15)\\ \mathbf{Y}_k &= \hat{\mathbf{T}}_k \mathbf{X}^* && (16)\end{aligned}$$

where $*$ denotes the complex conjugate and boldface printing denotes the representation in the spectral domain. The k-th elements of all feature vectors $y_k(\vec{s})$ at all locations $\vec{s}$ are given by the inverse Fourier transform of $\mathbf{Y}_k$. After calculating all $\vec{y}_k$ planes in the region of interest all the feature vectors are known at all locations in this region of interest. In Figure 4 this is graphically illustrated. Note the difference with Figure 2. All the

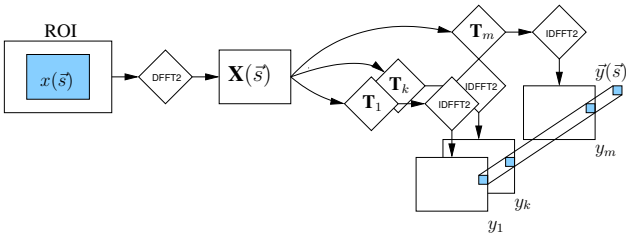

Fig. 4. Feature extraction in the spectral domain.

elements of $\vec{y}(\vec{s})$ are calculated for all locations with one multiplication per element.

The spectral correlation kernels, $\hat{\mathbf{T}}_k$, can be pre-calculated during training thus keeping the number of calculations minimal.

In Appendix C the computational complexity of the frequency domain implementation is compared to the Viola and Jones implementation, which is known for its efficiency and speed. The complexities of MLLL and VJ are not essentially different.

III. BILBO

The landmarks are disturbed by two types of errors: noise and outliers. The noise refers smaller errors and will be present in every estimate. If a sufficient number of landmarks is used, the effect of noise on the registration will be limited [8]. The outliers are the larger errors, which will seriously distort the registration. In order to reduce these larger errors, we present an outlier detection and correction method named BILBO. Although we assumed the landmarks to be independent for the derivation of MLLL in Section II, we will now explicitly use the dependence of the locations of the landmarks to correct outliers.

In related research fields subspace methods are used as an effective tool for removing noise from images. This has been done by, amongst others, Muresan and Parks [18], Goossens et al. [19] and Osowski et al. [20]. By keeping only the dominant features in the subspace and subsequently projecting back to the image space, the noise is reduced. Here we apply the same principle onto the shape. We define a subspace and BILBO projects the shape *there and back again* [21]

A. Theory

Correct shapes are assumed to lie in a subspace of $\mathbb{R}^{2d}$ with d the number of landmarks. Incorrect shapes, containing one or more erroneous landmarks, are assumed to be outside this subspace. Consider a measured shape $\vec{s}\,'$ that consists of a part $\vec{s}$ which fits the subspace $\mathbb{R}^n$ with $n < 2d$ and an error $\vec{\epsilon}$ which cannot be represented in this subspace.

$$\vec{s}\,' = \vec{s} + \vec{\epsilon} \tag{17}$$

Erroneous landmarks correspond to a pair of large elements, ϵ_i, of $\vec{\epsilon}$. BILBO aims to find those landmarks and correct them. We can estimate the error on the measured shape $\vec{s}\,'$ by

$$\vec{\epsilon} = \vec{s}\,' - \left(BB^T(\vec{s}\,' - \vec{\mu}_s) + \vec{\mu}_s \right) \tag{18}$$

Large elements of $\vec{\epsilon}$ indicates an outlier. If for a certain landmark the error is above a threshold, τ, its location is replaced with the location after projection.

$$\vec{s}\,'_i = \vec{s}_i \ \forall i \big| |\vec{\epsilon}_i| > \tau \tag{19}$$

This procedure is repeated until convergence has been reached, which is usually already after 1 iteration.

Training BILBO is done by finding the largest variations for all normalized training shapes. Normalised means that the shapes are aligned to a reference shape. The reference shape, which is the average shape when the found face coordinates have been scaled between 0 and 1. Our implementation is explained in Appendix B1.

Applying BILBO is schematically shown in Figure 5. It shows how the error, $\vec{\epsilon}$, is calculated. The error is used to determine which landmarks seem to be wrong and need to be corrected. This is done repetitively until all $|\vec{\epsilon}_i|$ are below the adaptive threshold τ. In Appendix B2 this will be discussed in more detail.

Though simpler, BILBO shows a resemblance to the Ransac algorithm [22] where also a distinction between "inliers" and "outliers" is made.

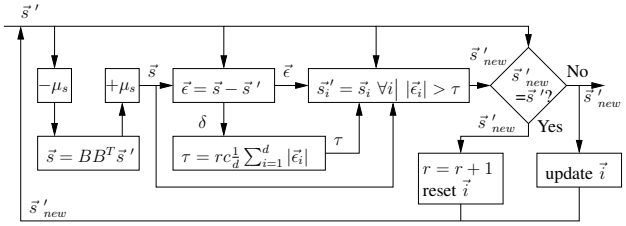

Fig. 5. A schematic overview of BILBO. The vector $\vec{i}$ keeps track of the landmarks to be updated. A detailed description can be found in Appendix B2.

IV. The Repetition Of Landmark Locating

The training images have been registered to a standard scale and pose before extracting the transformation matrix T and the parameters of Equation 9. Therefore, these do not fully model the orientation variations that occur in the images when landmarking. Because of this, MLLL would perform best on registered faces. This is, of course, normally impossible as landmarking is one of the steps of registration. We therefore propose to iterate the landmarking procedure. This procedure will be called: The Repetition Of Landmark Locating (TROLL). Once landmark candidates have been found, the image is registered and the landmarking is repeated on the registered image. We use MLLL, in combination with BILBO as the landmarking method, but other landmarking methods could also be used iteratively in the same manner. The processing time is linear with the number of iterations. We will choose the number of iteration such that further iterations yield no significant improvement.

V. Training and tuning

In this section we will discuss the training and tuning of the parameters of MLLL, BILBO and TROLL. The performance of these algorithms has a strong relation to the choice of the parameters.

First, we discuss the databases used in Section V-A. Second, this section will focus on tuning of the various parameters and their influence on the algorithms. An overview of these parameters and their final values is given in Table I. Repeatedly one parameter was optimized while all others were kept fixed until a stable solution was reached. We present only the results of the final parameter settings.

In order to evaluate the performance of the methods used we used the same error measure as Cristinacce [23]. The error measure, m_e is the mean euclidean distance between the landmarks and the manually labelled groundtruth coordinates as a percentage of the interocular distance Δ_{ocl}.

$$m_e = \frac{1}{n\Delta_{\text{ocl}}} \sum_{i=1}^{n} \sqrt{\delta_{i,x}^2 + \delta_{i,y}^2} \tag{20}$$

All results in this section are obtained by landmarking images in the training set. The final results obtained with the fully tuned algorithm on the testing sets are given in Section VI.

Sometimes the full parameter space was not explored but only the part where an optimum could be expected because exploration of the full parameter space is not feasible due to time constraints. Although the authors made an effort in finding a good solution it may, therefore, be a local optimum.

A. Databases used

We used two databases from which we drew several datasets for the experiments. Both the FRGC 2.0 [3] and the BioID [2] are publicly available. For testing we only used images in which the face was found by an unsupervised face finder, in this case the Viola and Jones [24] classifier from the OpenCV library with the "frontalface_alt2" cascade [25].

The BioID database consists of 1521 images, taken from 22 persons, which vary in pose, scale and illumination conditions, but which are mainly frontal. All images

TABLE I
OVERVIEW OF THE TUNING PARAMETERS AND CHOSEN VALUES.

Parameter	final value
MLLL	
Face size	384 [px]
Template size Nose ($n = v \times u$)	48x64 [px]
Template size Eye ($n = v \times u$)	64x96 [px]
Template size Mouth ($n = v \times u$)	64x96 [px]
Relative distance to the landmark	25 [%]
ARSVD blocksize (b)	500
Number of features (m)	219
Explained variance Landmark	81 [%]
Explained variance Background	100 [%]
Explained variance Total	98 [%]
BILBO	
Maximum number of iterations (r)	3
Minimal threshold (τ_{min})	0.055
Error weight (c)	1.15
Number of features in subspace	1
TROLL	
Number of repetitions	3

have been landmarked manually. The Viola-Jones face detector found a face in 1459 of the images (95.9%).

In total, the FRGC 2.0 database contains 39328 images, roughly one third of which are low quality images (LQ) and two third are high quality images (HQ). The FRGC 2.0 comes with hand labelled ground truth locations for four landmarks: the eyes, nose and mouth. We split the FRGC into a training set and a testing set: a training set containing 19674 images with subject ID number 4519 or lower and a testing set containing 19427 (98.8%) found faces in the 19654 images with subject ID numbers 4520 or higher. Both sets contain images from HQ and LQ.

B. Tuning MLLL

The MLLL has many parameters to tune. In Table I an overview of these parameters is given. For all parameters we started with an educated guess. Repetitively one parameter was optimized while the others were kept fixed. This was done until for all parameters a final setting was found, based on the landmarking performance in terms of either speed of accuracy.

It was possible, by reusing intermediate results, to keep the training of the algorithm sufficiently fast. Testing the algorithm was however slow because it had to be redone for each new parameter choice. In order to limit the tuning time, the parameters were tuned by landmarking the first 2000 images of the FRGC training set. This limitation implies the risk of overtraining on the first 2000 images of the training set. Verification on the larger dataset showed that this did not happen. Finally, after all parameters have been optimized the error measure, m_e calculated over the first 2000 images of the FRGC training set is 4.06 and over the full set it is 3.89. The fact that over the full set the error is lower suggests that there has been no significant overtraining in the tuning of the parameters.

1) Image size and landmark region of interest size: Since larger images imply larger areas to scan, the predetermined upper bound was an image size of 384×384 pixels. Experiments showed that smaller images resulted in larger errors. Therefore, the image size was set to 384×384. Note that for computational reasons we chose not to use images larger than 384×384. Improvement might be possible here.

Experiments with the template sizes showed that landscape shaped templates yielded lower errors than square or portrait shaped templates. For the eyes a template size of 64×96 gave best results. For the nose and the mouth the maximum performance was reached with templates of 48×64.

2) Selection of landmark and background training samples: In order to create a good separation between the landmark samples and the background samples, the background training samples should not include landmark templates. In Figure 6 we illustrate how the centre of the background training sample must have a minimal distance to the centre of the landmark. The minimal distance is relative to the width and height of the image, resulting in elliptical regions from which the centres of the background samples are taken. Experiments showed that a distance larger than 0.2 gave significantly better results than smaller distances. To be on the safe side this parameter was set to 0.25. The ellipse denoting the maximum distance had the same radius as half the template size, resulting in an elliptical doughnut where the centres of the background training samples are taken from.

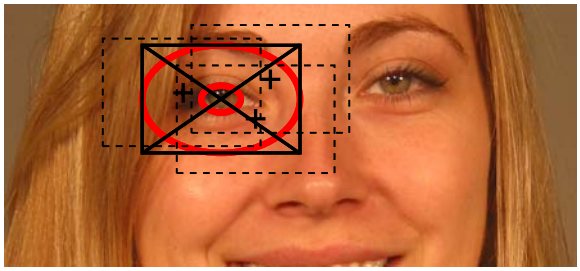

Fig. 6. Training sample selection. The landmark training sample is a rectangular region around the landmark, denoted with the solid rectangle and cross. Within this region a subregion is defined. This elliptic doughnut shaped area is the region where the centres of the background training samples, denoted by the pluses, are chosen from. Three examples are given as rectangles with a dashed border.

3) Block size: The block size in the ARSVD algorithm must be large enough to capture all the variation. It turned out that it is not a parameter with a very large influence on the final result as long as it is larger than 300. To be on the safe side we chose 500, as illustrated in Figure 7. For the HQ smaller block sizes would be allowed than for the LQ. In Table II the amount of kept variance for a block size is given for both Landmark and Background samples. In Figure 8 the amount of kept variance is illustrated for a blocksize of 500. It shows clearly that each time a block is added the variance within the blocks is modelled better. Finally near 100% of the variance in the new block is already modelled by the data.

4) Dimensionality reduction: MLLL has four parameters that determine the dimensionality reduction of the

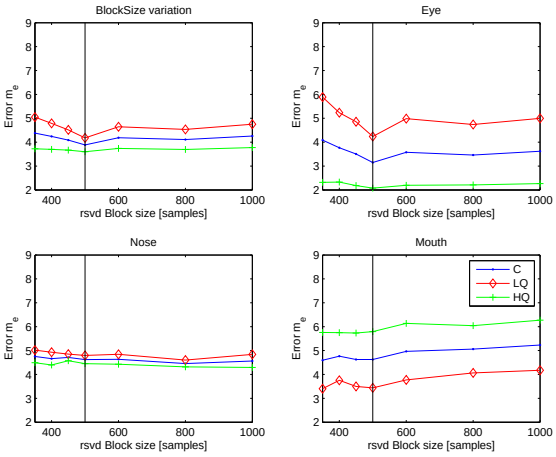

Fig. 7. The error m_e as function of the blocksize. Block sizes smaller then 300 will not result in enough features. It is clear that below 500 features the error grows when the number of features is reduced. More features do no improve the performance. The black line indicates the chosen value.

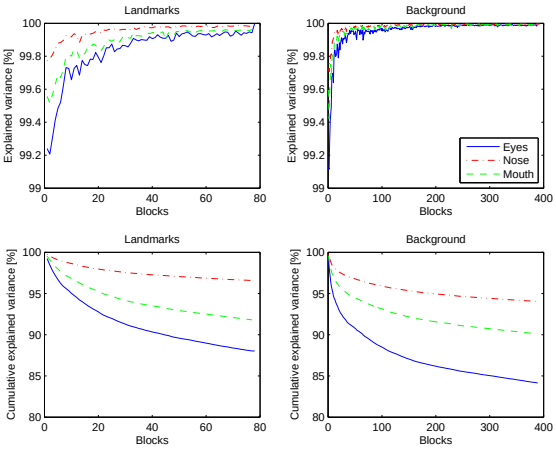

Fig. 8. The upper two graphs show the amount of variance which is kept after each feature reduction step. This goes to 100% when the data is modelled better and better. The lower graphs show the cumulative kept variance of the total data as a function of the number of processed blocks.

feature vector. The first two are the dimensionalities of the subspaces of the landmark and background data, cf. Equations 21 and 22 in Appendix A. The third parameter is the dimensionality of the joint subspace of background and landmark data, cf. Equations 23 to 27 in Appendix A. Instead of these dimensionalities, we will take the amount of variance retained in the, respective, subspaces as tuning parameters. The fourth parameter is the number of most discriminating features that is selected in the final feature reduction step. For every parameter is a trade-off between speed and accuracy. The chosen setting for each of these parameters has an impact on the others. Fewer features will give faster performance but too few will make the error m_e too large. Too many features will lead to overfitting, again resulting in poor performance. The choice of these parameters are discussed in the following paragraphs. In that procedure we start with an educated

TABLE II
AMOUNT OF KEPT VARIANCE USING A BLOCKSIZE OF 500 AND TRAINING ON ALL THE DATA OF THE FRGC TRAINING SET.

	Landmark	Background
Eye	88.0 [%]	84.2[%]
Nose	96.6 [%]	94.0[%]
Mouth	91.8 [%]	90.1[%]

guess and after that optimise the parameters one at a time, converging to a hopefully global optimum.

5) Explained variance landmark templates: Figure 9 shows that there is an optimum around 81% of kept variance, which is mainly due to a local minimum in the landmarking errors for the eyes. Errors for the eyes are the same for kept variances above 88% because the amount of kept variance due to the ARSVD is 88%.

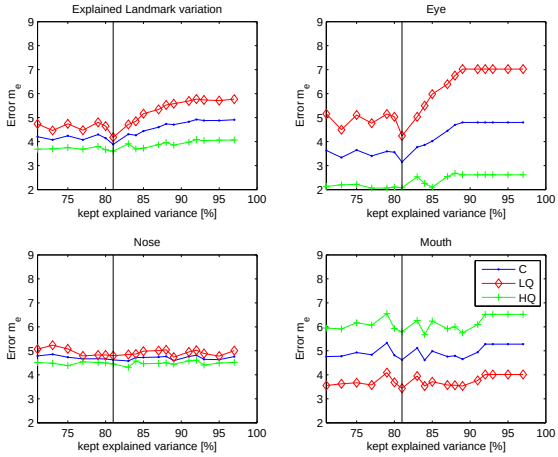

Fig. 9. The error, m_e, as function of the amount of explained landmark variance. The black line indicates the chosen value.

6) Explained variance background templates: There is not too much room to vary this parameter. The total amount of kept variance after the ARSVD is 94% for the eyes and even less for nose and mouth. Keeping 94% or more of all features means de facto keeping all features. The drop off is very steep because at 94% all 500 features are kept while going below 93.5% only few features are kept. Therefore this parameters is set to 100%, keeping all features in order not to limit the choice for the number of features m in Section V-B8.

7) Combined explained variance: As we can see in Figure 10 the influence of the overall explained variance is a rather limited. It is, apart from noiselike fluctuations, almost flat throughout its range. Important considerations for this parameter are computational speed during training and the fact that we want to keep enough features for the next phase to be effective. Nonetheless, we choose to tune our system to 98%, the local optimum.

8) Number of features during feature selection: The last feature selection step selects the number of features to be kept. As was explained in II-A1 the criterion here is the maximum of the quotient of the mean and the standard deviation. Figure 11 shows how the final selection of

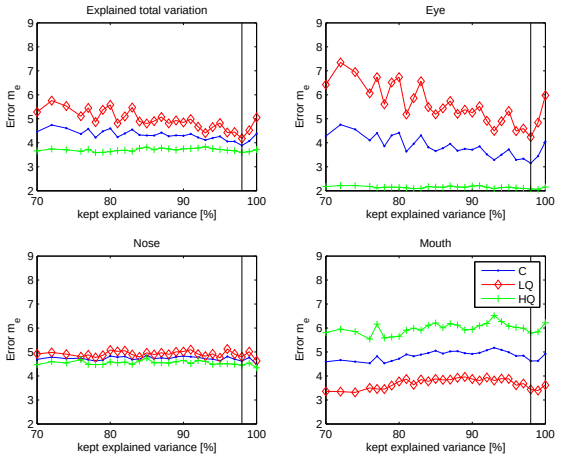

Fig. 10. The error, m_e, as function of the amount of total or overall explained variance. The black line indicates the chosen value.

features enables one to find a local optimum. Not all landmarks have a clear optimum. For the eyes it is clear that around 150 features is best. For both the nose and the mouth, above a certain value the error becomes more or less constant. The value of 219 was the overall best.

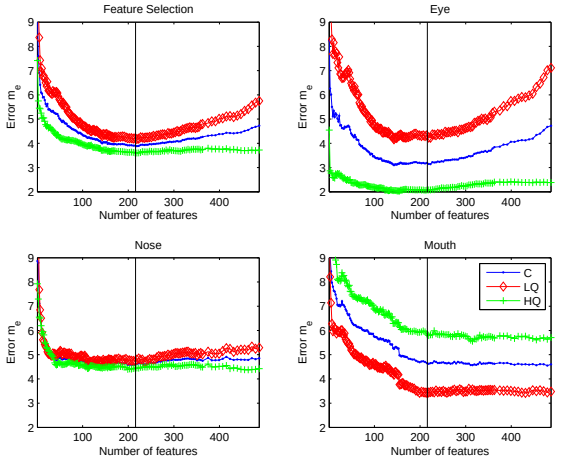

Fig. 11. The error, m_e, as function of the amount of total or overall explained variance. The black line indicated the chosen value.

9) Discussion: Interestingly, the m_e of 3.1 for the mouth on the LQ images is lower than the m_e of 5.8 for the HQ images. This is against the intuition that the error on HQ images should be lower. If we however calculate the errors for the full data set this effect disappears, as we would expect. The HQ error is 3.7 and the LQ error is 4.3. We, therefore, consider this to be a data anomaly.

C. BILBO

The BILBO outlier correction algorithm has four parameters to tune. The number of iterations, the minimal threshold, the weight factor and the number of features that are kept. Since the FRGC database has ground truth coordinates for four landmarks BILBO uses eight input features. In Figure 12 the first three modes of variation in the subspace are visualised in shape space. Experiments showed that by keeping only the first feature in the subspace the best results were obtained. The number of iterations was set to 3 because convergence was reached at that value for all the shapes in the training data. The final two parameters, the minimal threshold and the weight factor, were both optimized. The results are shown in Figure 13. The minimum is found for a minimal threshold, τ_{min} of 0.055 and an error weight c of 1.15. The mesh denotes the m_e without any outlier correction of 4.1% for reference purposes.

Examples of both correct and erroneous outlier corrections are given in Figure 14.

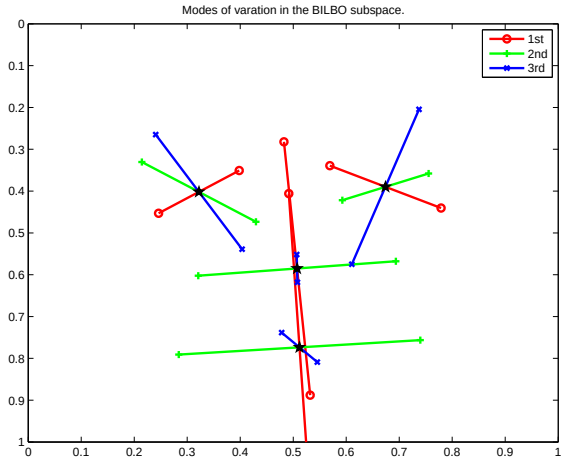

Fig. 12. The three modes with the highest variation in the BILBO subspace.

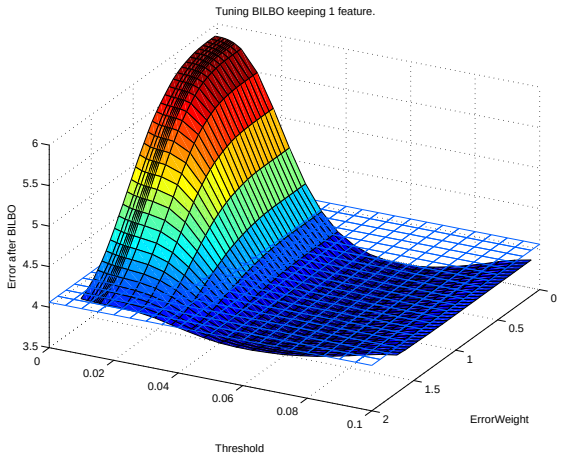

Fig. 13. The error, m_e as function of both the minimal threshold τ_{min} and the error weight c. The surface indicates the error when using BILBO. The mesh denoted the error without applying BILBO for reference purposes.

D. The Repetition Of Landmark Locating

The number of iterations determines how often we rerun the landmarker. Here that is MLLL in combination with BILBO. The choice of the number of iterations will

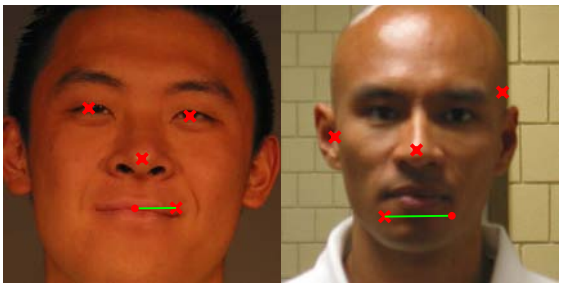

Fig. 14. Landmark outlier correction. The crosses denote the landmark location by MLLL while the dots denote the corrected location. In the left image the successful detection and correction of an outlier is shown. The right image shows an example where the input data is so bad that BILBO is unable to do anything meaningful.

be based on a trade off between accuracy, landmarking error and processing time. Since this parameter is linear with the total time needed we want to keep it as low as possible. In Table III it can be seen that with each iteration the error reduces, but not significantly after the 2nd iteration.

TABLE III
THE m_e FOR ALL LANDMARKS FOR FIVE ITERATIONS. CHANGES BEYOND THE SECOND ITERATION ARE NOT SIGNIFICANT. BOLDFACE DENOTES THE MINIMAL VALUE.

Landmark	1st	2nd	3rd	4th	5th
Combined	3.8	3.5	3.5	**3.4**	3.5
Eyes	3.2	3.2	**3.1**	**3.1**	**3.1**
Nose	4.5	**4.1**	**4.1**	**4.1**	**4.1**
Mouth	4.4	3.6	3.6	**3.5**	**3.5**

VI. FINAL RESULTS

In this section the results of the landmarking experiments are presented and discussed. All tuning parameters are set to values as found in Section V-B and given in Table I. In all experiments we distinguish between the high quality images (HQ), the low quality images (LQ) and the combined results (C). More information on the datasets has been given in Section V.

We present the results for three combinations: MLLL, MLLL+BILBO, and TROLL, which iterates MLLL+BILBO. Also we provide the results of two reference algorithms.

A. Reference algorithms

For reference purposes we provide two basic algorithms. The first returns the a priori landmarks given the face location and size as found by the Viola and Jones face detector. It will be denoted as the a priori landmark locator. The second algorithm is the OpenCV [25] implementation of the Viola and Jones face finder, but now trained for finding landmarks on the same datasets as MLLL [26].

B. Results

The results of all experiments are given in Table IV. With a few exceptions it can be said that both BILBO and TROLL improve the performance of MLLL. On the eyes the Viola and Jones landmark locator performs better on the LQ images and MLLL run on the HQ images. In general all methods perform better on the HQ images than on the LQ images. Virtually all methods perform better than the a priori landmark locator. Cumulative error plots for both the HQ and the LQ are given in Figures 15 and 16. In the latter case it can clearly be seen that for the eyes the Viola and Jones implementation outperforms all other methods, while on the mouth it lacks performance. Comparing the results for HQ and LQ shows that for the eyes the difference is large but at the same time for the nose and the mouth it is a lot smaller.

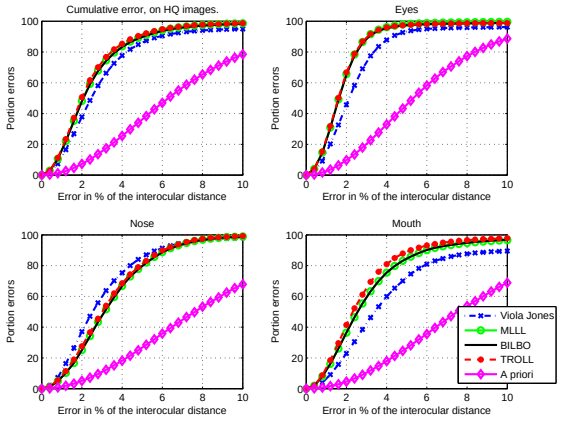

Fig. 15. Cumulative error distribution. Landmarkers trained on the FRGC training set. Testing on HQ of the FRGC testing set.

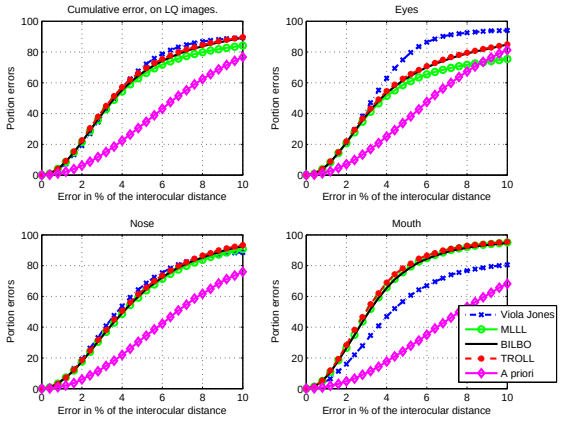

Fig. 16. Cumulative error distribution. Landmarkers trained on the FRGC training set. Testing on LQ of the FRGC testing set.

C. Discussion

1) MLLL: It is remarkable that for both nose and mouth there is a rather small difference between the HQ and the LQ. For the nose the LQ error is 1.2 times larger than the HQ error. For the mouth this is 1.4 times. On the

TABLE IV
THE m_e FOR ALL METHODS. THE RESULTS FOR MLLL, MLLL+BILBO AND TROLL ARE SHOWN. AS WELL AS TWO REFERENCE METHODS.

	Combined			Eyes			Nose			Mouth		
Training set: FRGC training set, Testing set: FRGC testing set												
	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ
A priori	7.3	7.2	7.6	6.2	5.9	7.0	8.2	8.5	7.5	8.5	8.4	8.8
Viola Jones	4.2	3.5	5.6	**2.9**	2.4	**3.9**	4.4	3.5	6.1	6.6	5.8	8.3
MLLL	3.9	2.7	6.3	3.8	**1.9**	7.5	4.3	3.6	5.6	3.8	3.4	4.5
BILBO	3.5	2.7	5.0	3.2	**1.9**	5.4	4.1	3.6	5.0	3.6	3.3	4.3
TROLL	**3.3**	**2.5**	**4.9**	3.1	**1.9**	5.4	**3.9**	**3.4**	**4.8**	**3.3**	**2.9**	**4.0**
Training set: FRGC training set, Testing BioID												
A priori	10.6			8.6			13.3			11.9		
Viola Jones	9.0			6.7			11.8			11.3		
MLLL	7.5			5.7			10.4			8.3		
BILBO	6.6			**5.3**			9.0			6.9		
TROLL	**6.3**			**5.3**			**8.1**			**6.6**		
Training set: BioID, Testing set: FRGC testing set												
	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ	C	HQ	LQ
A priori	8.3	8.4	8.2	7.7	7.6	7.9	8.4	8.7	**7.9**	9.3	9.6	8.9
Viola Jones	7.7	6.4	10.3	3.8	3.4	**4.7**	13.3	10.1	19.9	9.1	8.2	10.9
MLLL	6.9	6.0	8.6	3.3	2.5	4.8	12.5	13.1	11.5	8.5	6.0	13.4
BILBO	5.6	4.9	**6.9**	3.4	2.6	4.9	8.5	8.2	9.2	7.0	6.1	8.5
TROLL	**5.3**	**4.4**	7.0	**3.3**	**2.4**	4.9	**8.0**	**7.1**	9.6	**6.8**	**5.8**	**8.7**

contrary the eyes show a big difference with a 2.8 times larger error for the LQ data.

The weakest performance of MLLL is on the LQ eyes when trained on the FRGC training set. We suspect several causes of this. First of all, the illumination conditions which severely darken the eyes. Also the camera is sometimes out of focus. In the LQ images some people wear glasses, sometimes with a glare on it. Finally, people sometimes turn their eyes aside or close their eyes at the moment the image is taken. In Figure 17 some examples are shown. From these it can be seen that these causes affect the nose and mouth to a lesser degree than the eyes. This is supported by the fact that MLLL performs much better on the LQ data when trained on the BioID database, which does not contain such deteriorated samples. It is also true that for images in the testing set with the imperfections as shown in Figure 17, MLLL makes the worst errors. Having poor quality images in the training set apparently does not make MLLL more robust.

Fig. 17. Examples of LQ training samples that, for the eyes, deteriorate the landmarkers. Clockwise from the upper left we have illumination, illumination in combination with focusing on the background, looking sideways and finally glasses with glare on them. Having these in the training set does not improve the performance.

2) BILBO: The effect of BILBO can be analysed in more detail than just as the reduction of the error m_e after MLLL. In Figure 18 the change of the error per image are shown as the blue solid line. For illustrative purposes the errors are sorted by the improvement by BILBO. On the left negative improvements represent the images where the estimates of the landmark coordinates had been deteriorated. Moving to the right it is clear that most of the images are not changed at all. Finally on the right the improvements are shown. The area between the blue solid line and the null-line is a measure for the total improvement. For the low quality images the positive improvement by BILBO is eleven times the deterioration. For the high quality images the effect is only just positive (1.3 times). The more detailed information in Table IV shows that BILBO improves the results for all landmarks and datasets with the exceptions of the HQ images of the eyes when training on the FRGC training set and testing on the FRGC testing set. This is however only a very small effect.

3) TROLL: For the nose and the mouth TROLL yields the best results. The improvement caused by TROLL is analysed in the same way as the improvement of BILBO. This is also illustrated in Figure 18. Analogous to BILBO the gain is highest on the low quality images, namely 6 times. For the high quality images the improvement is a factor of 1.7. In contrast to BILBO there is a smooth transition from deterioration to improvement without a

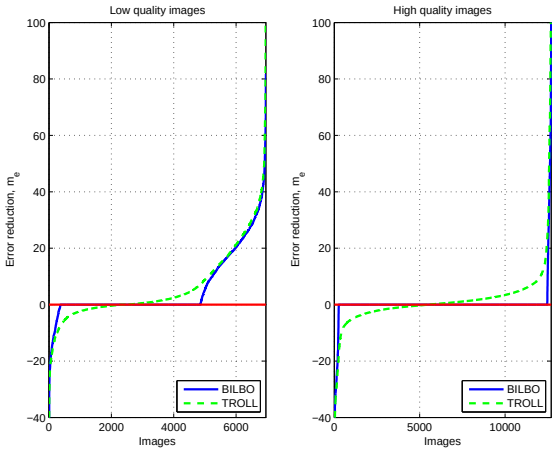

Fig. 18. The error reduction by BILBO and TROLL, sorted by the improvement. The blue line denotes the error reduction by BILBO. The green dashed line denotes TROLL. Negative values show a deterioration of the results and positive values an improvement.

dead zone where the coordinates are not adjusted.

It proved that TROLL was not able to get any intelligible results if the initial face bounding box had dimensions so that some landmarks fall outside the search areas. This would cause MLLL in the first run to give just any random position, and thus TROLL can drift away. An example is given in Figure 19. Because the face finder found the face on the wrong scale, the nose and mouth are not within the search regions, denoted by the red rectangles. The results of MLLL, BILBO and TROLL are thus not meaningful. In the FRGC testing set there are 810 images for which one of the landmarks is not in the search area. The impact on the overall performance is limited: it increases the error measure roughly 0.1%.

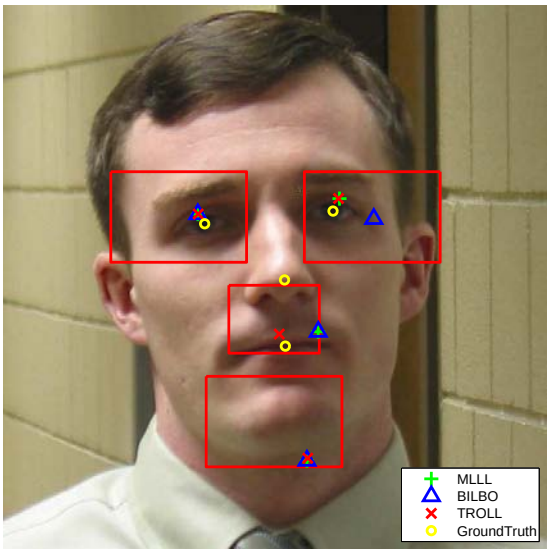

Fig. 19. Poor performance of all algorithms because the face finder found the face on the wrong scale. The landmarks lie outside the search areas denoted by the red rectangles.

TABLE V
COMPARING OTHER WORK ON THE EYES. BOLDFACE DENOTES THE MINIMUM. ITALICS DENOTES AN ESTIMATE NOT PROVIDED BY THE AUTHORS.

	Combined	HQ	LQ
Wang et al. [6]	**2.67**		
Campadelli et al. [7]	*2.7*	2.65	**2.88**
Viola and Jones [10]	2.9	2.4	3.9
Troll	3.1	**1.9**	5.4

4) Comparison to other work: Several papers report results on eye-finders. Unfortunately the authors were not able to find any work for nose and mouth localization that could be compared on the FRGC database. Here we only focus on the ones that report results on the eyes and the FRGC for ease of comparison.

There is a difference between the shape Shape Optimised Search (SOS) by Cristinacce et al. and our proposed methods BILBO: SOS is an integral part of the approach and BILBO is performed as an outlier correction method after landmarking.

Wang et al. [6] used Adaboost in combination with multiple weak probabilistic classifiers. Using non FRGC training data from multiple sources they report a mean Euclidian distance error on the eyes of 2.67% of the interocular distance on the FRGC 1.0 database, which is a subset of the FRGC 2.0 database. Their results can be compared to ours because they tested on the FRGC 1.0. The FRGC 2.0 database is larger but includes the FRGC 1.0 database. Wang et al. seem to have a similar, but slightly better result on the eyes than the Viola and Jones algorithm which has an m_e of 2.9 for the Viola and Jones method and a 3.1 for TROLL.

Campadelli et al, [7] used a combination of Haar classifiers and Support Vector Machines. They report a 2.65% error on the HQ data and a 3.88% error on the LQ data of the FRGC 1.0 database. These results are also similar to the ones we obtained with a Viola and Jones detector. The MLLL performs significantly better on the HQ data while on the LQ data it is worse. These results are summarised in Table V.

In previous work by the authors [27] results for earlier versions of MLLL, which were not tuned nor optimized, and BILBO were given. See Table VI. These versions were trained on the BioID database and tested on the FRGC 1.0 database. The new results are significantly better for MLLL. For newly trained BILBO the results on the mouth and the nose yield slightly higher errors. This can be explained by the fact that BILBO used 4 landmarks while the 'old BILBO' in [27] used 17 and therefore could make better use of the dependency of the landmarks. Note that MLLL and BILBO were tuned using the FRGC 2.0 database. The tuned parameters were not changed when training on the BioID database. Therefore we do not have optimal performance when training on the BioID database. The numbers are given in Table VI. This shows that tuning can lead to significantly better result for MLLL. Also it shows that BILBO using more landmarks is useful for BILBO.

TABLE VI
COMPARING MLLL AND BILBO TO OLDER WORK. BOLDFACE DENOTES THE MINIMUM. TRAINED ON THE BIOID DATABASE, TESTED ON THE FRGC. IT SHOULD BE NOTED THAT THE OLD VERSIONS WERE TESTED ON THE FRGC VERSION 1 DATABASE WHILE THE NEW ONES WERE TESTED ON OUR TESTING SET OF THE FRGC VERSION 2.

	Combined	Eyes	Nose	Mouth
old MLLL	10.3	6.2	17.1	7.7
new MLLL	6.9	3.3	12.5	8.5
old BILBO	6.2	5.4	8.0	**5.6**
new BILBO	5.6	3.4	8.5	7.0
TROLL	**5.3**	**3.3**	**8.0**	6.8

The MLLL method presented here used one set of parameters to find eyes, nose and mouth. These parameters have not been optimized for finding the eyes as was the case with the methods we used for comparison. Seeing that these specifically-for-the-eyes-trained locators perform similarly we are confident to say that our results have a good probability of performing better when tuned separately for each landmark. Finally, all methods are coming into the accuracy of the manual landmarks. The manual groundtruth landmarks are sometimes, according to the authors, with larger error then the proposed methods. Figure 20 provides some examples. Here we see that the manual landmarks of the nose are not placed consistently, at least for these examples. Unfortunately the accuracy of the manual landmarks is unknown. The manual landmarks are given as natural, rounded, numbers. Locally assuming a uniform distribution for the real locations the quantisation error can be calculated to be in the order of 0.4 pixels. This corresponds to a m_e in the order of 0.2%. This is less than one tenth of the mean error and therefore not likely to significantly enlarge the errors.

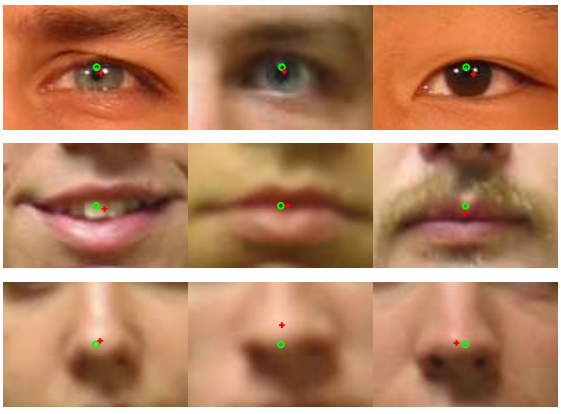

Fig. 20. This figure provides some examples where the landmarkers MLLL, BILBO and TROLL give an equal or better estimates than the manual landmarks. The green circle denotes the manual position and the red cross denotes the position found by TROLL.

5) Recommendations: For both training databases MLLL, BILBO and TROLL are trained using the same tuning parameters. Optimising for each landmark will surely improve the results because the current setting is probably a local optimum for minimizing for all landmarks at once. In the same fashion we treated the HQ and the LQ data equally. If we would have optimized MLLL for HQ and LQ and each landmark separately, the results are likely to improve.

In Section II we assumed the landmarks to be independent. This assumption is known to give a simplification of the truth. Not doing this very likely will improve the accuracy and robustness further because using this dependence in hindsight, as BILBO does, already improves the results.

VII. CONCLUSION

We presented several specific landmarking methods. The MLLL is based on Bayesian classifiers and is presented with a new theoretical framework based on maximum a posteriori. Two important extensions are proposed. BILBO is an outlier correction method and TROLL an iterative implementation of the combination of MLLL with BILBO. We show that all methods perform comparable to methods proposed by others, even though we present a more general implementation whereas others present a landmarker specifically for the eyes. TROLL has an overall error m_e of 3.3% of the interocular distance, which is far better than results obtained with earlier versions of MLLL. This shows that training on more data, as well as tuning the parameters, is worthwhile. BILBO also proved to be a useful tool, even if operated on only 4 landmarks. Iterative implementation of MLLL and BILBO proved to be a further improvement of the results significantly. TROLL shows the best overall performance of the presented algorithms. Although the setting of this paper is landmarking on facial images the algorithms can be applied to many landmark versus background classification problems in images.

It is to be expected that the results for the individual landmarks can be further improved by parameter tuning for each landmark individually. The same is true for training separately on the HQ of LQ data.

In Section II we assumed the landmarks to be independent. This assumption is known to be a simplification of the truth. Dropping this assumption very likely will improve the accuracy and robustness further, because using this dependence in hindsight, as BILBO does, has already shown to improve the results.

Two solutions to implementation issues are presented, namely the ARSVD and a spectral template matcher. The first makes it possible to do a singular value decomposition on large data with sufficient accuracy. The latter speeds up the execution of MLLL tenfold. Both were essential for final performance in terms of speed, accuracy and the possibility to investigate the parameter space while tuning.

Finally, because the accuracy of the manual groundtruth data the quality of current state of the art landmarkers is difficult to calculate reliably and difficult to compare. Even though this might pose a problem in evaluating the quality of the landmarkers this should not limit the ambition to improve them.

APPENDIX

A. MLLL

Here we briefly list the steps in the algorithms for the dimensionality reduction and the whitening of the data.

1) Dimensionality reduction: The subspace should contain a good representation of both the landmark data, X_l, and the background data, X_b.

i. Create the data matrices X_l and X_b where each column is a single training sample $\vec{x}(\vec{s})$.

ii. Calculate a basis of both landmark and background data:

$$U_{[l,b]}S_{[l,b]}V_{[l,b]}^T = (X_{[l,b]} - \mathrm{M}_{[l,b]}), \quad (21)$$

where $\mathrm{M}_{[l,b]} = \vec{\mu}_{[l,b]}[1 \dots 1]$, ie. a matrix whose columns are the column average of X. The subscript $[l, b]$ denotes that it applies to both the landmark and background data.

iii. For computational reasons only the first columns of U_b and U_l, which contain a fixed amount of the variance are kept.

$$\hat{U}_{[l,b]} = [\vec{u}_{[l,b],1}\vec{u}_{[l,b],2} \dots \vec{u}_{[l,b],n_l}], \quad (22)$$

where n_l and n_b denote the number of columns kept. Note that $\hat{U}_l$ and $\hat{U}_b$ are not mutually orthogonal.

iv. The orthonormal basis should also contain the difference vector between both means. Therefore we estimate the normalised average landmark projection $\vec{u}_l$. This is the difference between the two landmark means, normalised to unity length.

$$\vec{u}_{lb} = \frac{\vec{\mu}_l - \vec{\mu}_b}{|\vec{\mu}_l - \vec{\mu}_b|}. \quad (23)$$

v. Transform the combined matrix $[\hat{U}_l\ \hat{U}_b]$ so that it is orthogonal to u_{lb}.

$$U_{lb} = (I - \vec{u}_{lb}\vec{u}_{lb}^T)[\hat{U}_l\ \hat{U}_b]. \quad (24)$$

vi. Make U'_{lb} an orthonormal basis of U_{lb}

$$U'_{lb}SV^T = U_{lb}. \quad (25)$$

vii. The final basis is given by

$$U = [\vec{u}_{lb}\ U'_{lb}]. \quad (26)$$

viii. For the third time reduce the number of features:

$$\hat{U} = [\vec{u}_1\vec{u}_2 \dots \vec{u}_j]. \quad (27)$$

ix. Project the data onto the subspace

$$X'_{[l,b]} = \hat{U}^T(X_{[l,b]} - M_b). \quad (28)$$

2) Whitening the data: Whitening the data is done so that both the covariance matrices are diagonal and the background data is unity in variance. This later enables simple computation of Equation 5 or its final implementation Equation 9.

i. It follows from Equation 28 that the mean of X'_b, M'_b is zero. Perform an SVD on X'_b:

$$U_w S_w V_w^T = X'_b. \quad (29)$$

ii. Transform the data so that the background variance is unity:

$$X''_{[l,b]} = \frac{S_w^{-1}U_w^T}{\sqrt{n_b}}X'_{[l,b]}. \quad (30)$$

where S_w and U_w follow from the SVD in Equation 29. After this tranform the backgound covariance matrix is (approximately) unity.

iii. Diagonalise the landmark covariance. The background covariance matrix remains unity. Perform an SVD on the transformed landmark data X''_l:

$$U_d S_d V_d^T = X''_l - \frac{S_w^{-1}U_w^T\hat{U}^T}{\sqrt{n_b}}(\mathrm{M}_l - \mathrm{M}_b). \quad (31)$$

iv. This results in a projection matrix U_d. The transformation from the original image space to the subspace, which renders the background covariance matrix (aproximately) unity and (approximately) diagonalizes the landmark covariance matrix, is now defined as:

$$T = \frac{U_d^T S_w^{-1}U_w^T\hat{U}^T}{\sqrt{n_b}}. \quad (32)$$

B. BILBO

1) Training: BILBO is trained on a set of shapes, taken from the groundtruth data, arranged as the columns of a matrix S. The training consists of the following steps:

i. All shapes normalised in scale so that the region where the VJ face finder found the face is between 0 and 1. Using this method we model the real distributions of the data. All coordinates in S are thus between 0 and 1.

ii. Perform a singular value decomposition $(S - \vec{\mu}_s) = BWV^T$, with $\vec{\mu}_s$ the mean shape.

iii. Reduce the dimensionality of the subspace by taking only the first $n < 2d$ columns of B.

2) Algorithm: To correct a shape the following algorithm is used:

i. Estimate the shape after transformation, $\vec{s} = BB^T\hat{s}$.

ii. Determine the Euclidean distance $|\vec{\epsilon_i}|$ per landmark between $\vec{s}$ and $\vec{s}\,'$.

iii. Determine the threshold

$$\tau = rc\frac{1}{d}\sum_{i=1}^{d}|\vec{\epsilon_i}|, \quad (33)$$

with c a constant and r the iteration number. Do not choose τ smaller then a predetermined threshold.

iv. For the landmarks of which $|\vec{\epsilon_i}| > \tau$, replace in $\vec{s}\,'$ by the corresponding coordinates from $\vec{s}$: $\vec{s_i}\,' = \vec{s_i} \forall i \big|\ |\vec{\epsilon_i}| > \tau$.

v. Repeat steps i to iv. Once for a landmark $|\vec{\epsilon_i}| < \tau$ stop updating it. Continue until all landmarks satisfy $|\vec{\epsilon_i}| < \tau$. Keep track of the coordinates which are allowed to change (update $\vec{i}$).

vi. Repeat step i to v changing all coordinates until stable or $r = 5$. Allow all landmark coordinates to update (reset $\vec{i}$).

vii. Transform the coordinates back to the original scale.

In Figure 5 a schematic overview of the shape correction algorithm is shown.

C. Complexity

1) MLLL: Consider a ROI containing n pixels. The number of operations per DFFT2 is then $O(n \log_2(n))$. After feature reduction the number of features is m. The number of DFFT2s to be computed is $m + 1$, as can be seen in Figure 4. Computing the likelihood ratio after feature computation, Equation 9, at every pixel location has a complexity of $O(5mn)$. Number of operations per ROI for finding the maximum value is $O(n)$. This makes the total number of operations per ROI:

$$(m + 1)O(n \log_2(n)) + nO(5m) + O(n) \tag{34}$$

Dividing by n gives the number of operations per pixel in ROI

$$O(m(\log_2(n) + 6)) \tag{35}$$

The large ROIs used are 256×256 pixels, which means that $n = 25088$. We used $m = 219$ features. Equation 35 results in a complexity of $O(5000)$ operations per pixel in the ROI.

2) Viola and Jones: The complexity of the Viola and Jones algortihm depends on the numbers of scales S, cascades C, and features K. Estimates for these numbers are taken from [26]; $S = 11$, $C = 15$, $K = 30$, on average. The total number of operations per pixel in the ROI are upperbounded by $O(S \times C \times K) \approx O(5000)$.

REFERENCES

[1] NIST, "Face recognition vendor test, 2006," http://www.frvt.org/.

[2] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust Face Detection Using the Hausdorff Distance," in *Audio- and Video-Based Person Authentication - AVBPA 2001*, ser. Lecture Notes in Computer Science, J. Bigun and F. Smeraldi, Eds., vol. 2091. Halmstad, Sweden: Springer, 2001, pp. 90–95. [Online]. Available: citeseer.ist.psu.edu/article/jesorsky01robust.html

[3] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[4] T. Riopka and T. Boult, "The eyes have it," in *Proceedings of ACM SIGMM Multimedia Biometrics Methods and Applications Workshop.*, Berkeley, CA, 2003, pp. 9–16.

[5] D. Cristinacce, T. Cootes, and I. Scott, "A multi-stage approach to facial feature detection," in 15^{th} *British Machine Vision Conference, London, England*, 2004, pp. 277–286.

[6] P. Wang, M. B. Green, Q. Ji, and J. Wayman, "Automatic eye detection and its validation," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. Washington, DC, USA: IEEE Computer Society, 2005, p. 164.

[7] P. Campadelli, R. Lanzarotti, and G. Lipori, "Precise eye localization through a general-to-specific model definition," in *British Machine Vision Conference (BMVC), Edinburgh, UK, 2006.* BMVA, 2006, pp. 187–196. [Online]. Available: http://hdl.handle.net/2434/24373

[8] G. M. Beumer, A. M.Bazen, and R. N. J. Veldhuis, "On the accuracy of EERs in face recognition and the importance of reliable registration." in *SPS 2005*. IEEE Benelux/DSP Valley, April 2005. [Online]. Available: http://acivs.org/sps2005/

[9] G. M. Beumer, Q. Tao, A. M. Bazen, and R. N. J. Veldhuis, "A landmark paper in face recognition," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on, Southampton, UK*. Los Alamitos: IEEE Computer Society Press, April 2006.

[10] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, 2002. [Online]. Available: citeseer.ist.psu.edu/viola01robust.html

[11] D. Cristinacce and T. Cootes, "A comparison of shape constrained facial feature detectors," in 6^{th} *International Conference on Automatic Face and Gesture Recognition 2004, Seoul, Korea*, 2004, pp. 375–380.

[12] M. Everingham and A. Zisserman, "Regression and classification approaches to eye localization in face images," in *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 441–448.

[13] A. M. Bazen, R. N. J. Veldhuis, and G. H. Croonen, "Likelihood ratio-based detection of facial features," in *Proc. ProRISC 2003, 14th Annual Workshop on Circuits, Systems and Signal Processing*, Veldhoven, The Netherlands, nov 2003, pp. 323–329.

[14] H. van Trees, *Detection, Estimation and Modulation Theory, Part I*. New York: John Wiley and Sons, 1968.

[15] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[16] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[17] R. C. Gonzales and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1977.

[18] D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," in *ICIP (1)*, 2003, pp. 101–104.

[19] B. Goossens, A. Pizurica, and W. Philips, "Noise removal from images by projecting onto bases of principal components." in *ACIVS*, ser. Lecture Notes in Computer Science, J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, Eds., vol. 4678. Springer, 2007, pp. 190–199.

[20] S. Osowski, A. Majkowski, and A. Cichocki, "Robust PCA neural networks for random noise reduction of the data," in *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97) -Volume 4*. Washington, DC, USA: IEEE Computer Society, 1997, p. 3397.

[21] J. Tolkien, *The Hobbit – There and back again*. London: George Allen and Unwin, 1937.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] D. Cristinacce and T. F. Cootes, "Facial feature detection and tracking with automatic template selection," in *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 429–434.

[24] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features." in *CVPR (1)*, 2001, pp. 511–518.

[25] Intel, "Open computer vision library," http://sourceforge.net/projects/opencvlibrary/.

[26] Q. Tao, "Face verification for mobile personal devices," Ph.D. dissertation, Univ. of Twente, February 2009.

[27] G. M. Beumer and R. N. J. Veldhuis, "A map approach to landmarking," in *Proceedings of the 28th Symposium on Information Theory in the Benelux*, Enschede, The Netherlands, May 24/25 2007, pp. 183–187.