

# Automatic Fitting of a Deformable Face Mask Using a Single Image

Annika Kuhl, Tele Tan, and Svetha Venkatesh

Department of Computing  
Curtin University of Technology  
GPO Box U1987, Perth, 6845, Western Australia  
(a.kuhl, t.tan, s.venkatesh)@curtin.edu.au

**Abstract.** We propose an automatic method for person-independent fitting of a deformable 3D face mask model under varying illumination conditions. Principle Component Analysis is utilised to build a face model which is then used within a particle filter based approach to fit the mask to the image. By subdividing a coarse mask and using a novel texture mapping technique, we further apply the 3D face model to fit into lower resolution images. The illumination invariance is achieved by representing each face as a combination of harmonic images within the weighting function of the particle filter. We demonstrate the performance of our approach on the IMM Face Database and the Extended Yale Face Database B and show that it outperforms the Active Shape Models approach [6].

**Key words:** Deformable Face Mask, Eigenfaces, PCA, ASM

## 1 Introduction

Accurately detecting faces and geometrically aligning key features of the face is an important pre-requisite for many face recognition algorithms. The appearances of faces can be affected by the rotation of the face (both in-plane and out-of-plane rotations), image resolution and lighting conditions which all make the face detection problem even more challenging. Although some of these factors may not be an issue in a controlled indoor environment (e.g. access control system), it become profoundly important when dealing with outdoor environment and conditions where CCTV cameras are becoming more prevalent. In this paper, we propose a solution to accurately localise a face by automatically aligning the key landmarks of the face with a 3D face model. The proposed technique is designed to cope with low image resolution and varying illumination conditions.

The problem of fitting a deformable mask to an image has previously been addressed by finding facial features first and fitting the face mask to these points afterwards [16]. Active Shape Models (ASM) are commonly used for detecting facial feature points. ASM is a statistical model of the shape of a class of object. During the model matching process the position of landmark points along the contour lines of the shape are optimised. First proposed by [6], the ASM has been extended by [19] using an hierarchical CONDENSATION (particle filter)

framework and by [24], where the shape estimation problem is formulated in a Bayesian framework.

While the ASM only optimises around the contour lines of the object, Active Appearance Models (AAM) [11] represent both shape and texture of a given class of object. During the training process a generative model is built such that both shape and texture are controlled by a set of parameters. AAM have been extended to fit faces in low resolution images [7]. The authors therefore incorporated the image formation process into the fitting criterion. In [22], AAMs were combined with 3D Morphable Models to allow for a three-dimensional fitting. Furthermore in [2], AAMs have been extended to incorporate lighting changes using the work of [3].

Our approach differs from AAMs in that the possible shape deformations are already given by the 3D mask mesh we are utilising. We do not attempt to learn them from a set of training images. By using a 3D face mask for fitting instead of a 2D AAM we also incorporate three-dimensional information about the shape and the pose of the face. Furthermore we do not aim to generate a 3D model of the face like in [23, 4, 14], since the utilised face mask is not flexible enough to adapt to slight differences in bone structure.

This work addresses the problem of automatically fitting a deformable face mask to a previously unseen near frontal image of a face and thus implicitly finding facial feature. We use Principle Component Analysis (PCA) and a novel texture mapping technique to build a face model which is then used by a particle filter. Once the model is build offline it can be applied to input image resolutions that differ from the resolution of the training images. We achieve lighting invariance by incorporating the work of [3] into the weighting function of the particle filter.

Accurate detection of facial feature points is a pre-processing step for a number of applications, like fitting a deformable model for tracking [18] or 3D face modelling and recognition [4]. While feature point detection is feasible in high resolution images of faces under neutral lighting conditions most approaches struggle in low resolution and changing lighting. We therefore propose a method that estimates the pose parameters as well as person-specific shape parameters to fit a deformable 3D face model to a previously unseen image, instead of detecting facial feature points and then fitting a mask. This approach is applicable in low resolution and in arbitrary lighting conditions.

The significance of the proposed method is (1) automatic fitting, (2) light invariance, (3) automatic face shape fitting. By automatically fitting a deformable face mask model to a near frontal face, important facial feature points are located implicitly. Thus avoiding laborious manual feature point selection or manual mask alignment. The work of [3] is incorporated into the weighting function of the particle filter in order to compensate for arbitrary illumination, which makes our approach more robust and increases its application. The particle filter is used to estimate the person specific shape parameters to fit a deformable face mask to previously unseen faces. There are no information about the person required, which again increases the robustness and applicability of our approach.

## 2 Method

Our proposed method first builds a face model by fitting a deformable mask to a set of different face images. Using PCA a set of principle components, the so called Eigenfaces, are then calculated from these textured masks. During runtime a particle filter is used to estimate the pose and the shape parameters of the mask given a single image of a new face. Illumination invariance is achieved by incorporating the work of [3] into the weighting function of the particle filter.

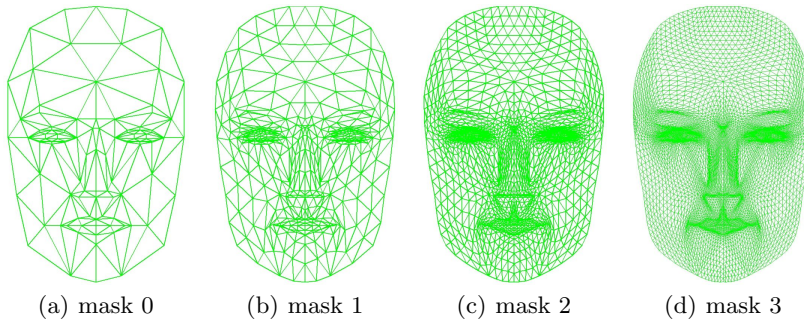
### 2.1 The Face Model

The offline step includes building a face model from a set of training images using principle component analysis (PCA) [20]. Instead of using vectorised 2D face images we fit a deformable mask to near frontal faces and apply PCA on a vector comprising the grey values of each mask triangle.

We utilise the CANDIDE-3 face model [1]. This 3D deformable face model consists of 104 vertices  $\mathbf{P}_i, i = 1, \dots, n$ , where  $\mathbf{P}_i$  are the 3D coordinates of the  $i^{\text{th}}$  vertex. These vertices form the complete mesh that consists of 184 triangles which are described by the vector  $\mathbf{g}$

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{S}\boldsymbol{\gamma} \quad (1)$$

where  $\bar{\mathbf{g}}$  is the neutral face and the columns of  $\mathbf{S}$  are the shape parameters that are controlled by  $\gamma_k \in [-1, \dots, 1]$ . Each shape parameter is a list of vertices and their displacement that result in the corresponding shape deformation. The original model allows for 14 different shape parameters; we reduce this number to 7, namely ‘eyebrows vertical position’, ‘eyes vertical position’, ‘eyes width’, ‘eye separation distance’, ‘nose vertical position’, ‘mouth vertical position’ and ‘mouth width’. This is done because we are mainly interested in the actual position of the facial features with respect to the face rather than the exact shape of the overall face. The number of parameters and the deformability of the mask do not allow for a precise estimation of a person-independent face shape.



**Fig. 1.** (a) Original CANDIDE-3 face mask, (b), (c) and (d) are subdivided masks after 1, 2 and 3 subdivision steps respectively. Source: [15] © 2008 IEEE.

This face mask is subdivided using the Modified Butterfly algorithm [26] three times to finally produce 5984 vertices and 11776 triangles. This subdivision

of the mask allows for different mask resolutions varying from only a few triangles to a very fine mask mesh when projected into the image. Fig. 1 shows the result.

After the mask is fitted to a particular face by adjusting the elements of  $\gamma$ , the middle of each mask triangle is projected into the image

$$\mathbf{I} = \mathcal{P}(\mathbf{g}, T) \quad \text{with } T = [T_{\text{int}}, T_{\text{ext}}] \quad (2)$$

where  $T_{\text{int}}$  are the intrinsic camera parameters determined using camera calibration techniques as in [25],  $T_{\text{ext}}$  are the extrinsic camera parameters, ie translation and rotation and  $\mathcal{P}$  the projection of the face mask model  $\mathbf{g}$  into the image using  $T$ . The resulting vector  $\mathbf{I}$  contains one grey value for each mask triangle. This differs from standard texture mapping techniques that usually require warping and interpolating the texture to fit the 3D model. This model has been applied previously for super-resolving faces in low resolution images [15]. By using a face mask model and a vector  $\mathbf{I}$  of concatenated grey values no geometric normalisation is required, the face masks of different persons are aligned implicitly.

These texture vectors of concatenated grey values  $\mathbf{I}$  are then used to calculate the average face  $\bar{\mathbf{x}}$  and a set of principle components  $\mathbf{X}$ , the so called Eigenfaces, using PCA [20]. A new face  $\mathbf{I}_{\text{new}}$  can then be represented as  $\hat{\mathbf{I}}$ , reconstructed from a combination of principle components  $\mathbf{X}$

$$\hat{\mathbf{I}} = \bar{\mathbf{x}} + \mathbf{X}\mathbf{X}^T(\mathbf{I}_{\text{new}} - \bar{\mathbf{x}}) \quad (3)$$

## 2.2 Light-invariant Automatic Head Pose and Shape Initialisation

After the face model is built offline it is used to fit the mask to new unseen faces. We therefore utilise the algorithm developed by [21] to detect near frontal faces, the first step of our approach. The image is then cropped according to the detection result. We initialise the face mask to fit this window by centralising the neutral face mask  $\bar{\mathbf{g}}$  such that it fills out the whole window.

Since every face might be cropped differently this first rough initialisation is further refined by a fast grid-search, to get a better approximation of the  $z$ -value, that controls the depth, ie size of the face mask. Starting from the initial  $z$ -value, we assign  $z$  with 14 different values in the range of  $\pm 5\%$  of the initial mask size and solve for the best  $x$  and  $y$  using a locally exhaustive and direct search as proposed by [9]. The error function  $e$ , that guides this search is the reconstruction error in face space

$$e(T_{\text{ext}}, \gamma) = \|\mathbf{I} - \hat{\mathbf{I}}\|_2 \quad (4)$$

with  $\gamma = 0$  and the rotation parameters within  $T_{\text{ext}}$  set to 0, since we assume a near frontal face. The set of pose parameters  $(x, y, z)$  that results in the smallest reconstruction error is then used for initialisation. To refine this initial pose a particle filter [10] is then used to converge to the correct pose ( $T_{\text{ext}}$ ) and person-specific shape parameters  $\gamma$  by ‘tracking’ repeatedly on a single image, essentially performing an incremental randomised search for the global maximum.

The reconstruction error in face space (Eq. 4) serves as the weighting function for the particle filter. Illumination invariance is achieved by incorporating the

work of [3]. They have proven that an image  $J$  can be expressed as a linear combination of so called harmonic images  $b_{nm}$

$$J_i = \sum_{n=0}^{\infty} \sum_{m=-n}^n \alpha_{nm} b_{nm}(p_i) \quad (5)$$

where  $\alpha_{nm}$  are the linear coefficients. Given a 3D model of the object, each harmonic image is a function of the albedo and the surface normal of each surface point  $p_i$ . Throughout this paper we use the first nine harmonic images [3]. Equation 5 is rewritten for simplicity as

$$J = \sum_{j=1}^9 \beta_j V_j(n, \rho) \quad (6)$$

where  $\beta_j$  is the  $j^{\text{th}}$  coefficient and the function  $V_j$  returns the  $i^{\text{th}}$  harmonic image given the surface normals  $n$  and the albedo  $\rho$  of all surface points of the 3D model.

### 2.3 Particle Filter Refinement

Particle filters are a statistical model commonly used in tracking [10]. They are based on the idea of Monte Carlo sampling to approximate the posterior probability distribution of a state – in our case the pose and shape parameters of the human face. This posterior is factorised as follows

$$P(x_t|y_{1:t}) \propto P(x_1) \prod_{t=1}^T P(y_t|x_t) \prod_{t=2}^T P(x_t|x_{t-1}) \quad (7)$$

where  $x_t$  is the state at time  $t$  (consisting of  $\{T_{ext}, \gamma_{1:K}\}$ ) and  $y_t$  is the observation at time  $t$ . For the bootstrap particle filter, Monte Carlo samples  $x_t^{(i)}$  can be drawn as follows [10]

$$x_t^{(i)} \sim P(x_t|x_{t-1}) \quad (8)$$

$$\tilde{w}_t^{(i)} = P(y_t|x_t), \quad w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_i \tilde{w}_t^{(i)}} \quad (9)$$

where  $\sim$  means “sample from” and  $w_t^{(i)}$  is the weight associated with particle  $i$ .

For this paper we set  $P(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1}, \Sigma)$ , that is we are sampling in the neighbourhood of the previous state.  $\Sigma$  is a diagonal covariance, ie the dimensions are sampled independently, with values set heuristically. In terms of standard deviation, 3D position is set to equal a shift of about one sixth of the face size, 3D rotation is  $15^\circ$  and the shape parameters  $\gamma_k$  are all set to 0.3.

These parameters are then used to deform the neutral face mask  $\bar{\mathbf{g}}$  according to Eq. 1 and project it into the image using Eq. 2. The face image vector  $\mathbf{I}$  is extracted according to Eq. 3 and the distance in feature space is then defined as

the norm of the difference between the retrieved image  $\mathbf{I}$  and the illumination-adjusted, reconstructed image  $\hat{\mathbf{I}}$ .

$$d(x_t) = \min_{\beta} \|\mathbf{I} - \beta V(n, \hat{\mathbf{I}})\|_2 \quad (10)$$

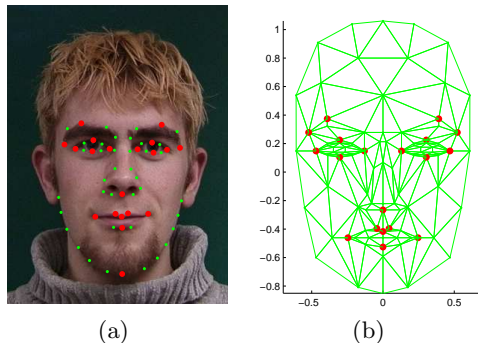
where  $n$  are the surface normals of the face mask that has been deformed by  $\gamma$  and the reconstructed face  $\hat{\mathbf{I}}$  is used as the reference albedo. Using this distance  $d$ , the weighting function  $P(y_t|x_t)$  is defined as a normalised vector

$$\tilde{w}_t^{(i)} = [\eta - d(x_t^{(i)})]^\lambda \quad (11)$$

where  $\eta = \max_i(d(x_t^{(i)}))$  and  $\lambda$  is an annealing factor to increase the spread of the particle weights [8], empirically set to  $\lambda=4$ . Note that we provide the particle filter with the *same* image at every  $t$ , which essentially performs an incremental refinement of the face mask fit rather than a tracking task. This utilises the genetic-algorithm-like nature of particle filters [8]. In order to ensure a convergence, we adjust  $\Sigma$  so that it declines as per  $\Sigma(t) = 0.8 \cdot \Sigma(t-1)$ .

### 3 Experiments

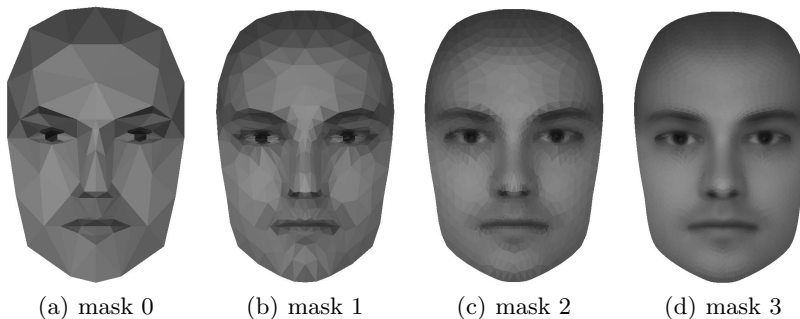
We use two face databases to perform our experiments, namely the the IMM Face Database [17] and the Extended Yale Face Database B [12]. The IMM Face Database consists of 240 images of 40 individuals, 6 images per individual, each exhibiting variations in pose, expression and lighting. We only use the first image (*‘full frontal face, neutral expression, diffuse light’*) and the fifth image (*‘full frontal face, neutral expression, spot light added at the person’s left side’*) of each person and we do not incorporate any prior knowledge on the lighting conditions in our model.



**Fig. 2.** (a) Sample image of the IMM face database with annotated facial landmarks, (b) CANDIDE-3 mask, red dots indicate point-to-point correspondences

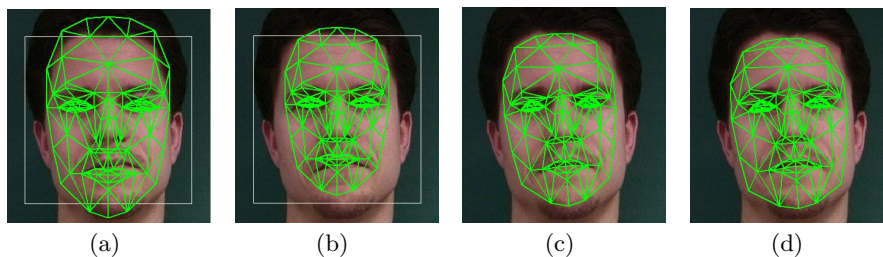
Each image of the IMM Face Database is labelled with 58 landmarks at various facial feature points, which can be used to automatically fit the CANDIDE-3 mask as a ground truth. We therefore assigned point-to-point-correspondences

between these landmarks and the corresponding mask vertices as shown in Fig. 2. The pose  $T_{ext}$  and shape parameters  $\gamma$  of the deformable face mask are estimated by utilising the Levenberg-Marquardt algorithm to minimise the Euclidean distance between the landmarks and the mask vertices. This mask fitting is performed on the first image of each of the 40 persons in the IMM Face Database which are used for training purpose as well as creating the ground truth.



**Fig. 3.** Mean face for each mask sized used after applying PCA to the first image of each of the 40 individuals of the IMM Face Database.

After the fitting process each mask triangle is projected onto the image and its grey value is extracted to form the face image vector  $\mathbf{I}$  for each of the 40 person. These vectors of concatenated grey values are then used to calculate the Eigenfaces as described in Section 2.1. We calculate the Eigenfaces and the mean face  $\bar{\mathbf{x}}$  for each mask size, varying from coarse to fine as shown in Fig. 3. We keep the top 70% of all eigenvectors, using the Matlab code provided by [5].



**Fig. 4.** (a) Result after face detection [white rectangle] and mask initialisation, (b) Result after refined initialisation using grid-search, (c) Final result of our approach, (d) Ground truth mask that was fitted to the labelled landmarks

Using this face model we conducted the following experiments on IMM Face Database: (1) testing on training, (2) leave one out, (3) testing on different lighting condition and (4) comparison with ASM. The progressive results of the proposed algorithm are shown in Fig. 4.

For experiment number one we used the first image of each person, the same that was used for calculating the face model, for fitting each of the four different

mask sizes, from coarse (mask 0) to fine (mask 3). Additionally we divided the initial image resolution of  $640 \times 480$  in half three times resulting in images of size  $320 \times 240$ ,  $160 \times 120$  and  $80 \times 60$  pixels, generating corresponding average face sizes of  $250 \times 160$ ,  $125 \times 80$ ,  $60 \times 40$  and  $30 \times 20$  pixels.

In the second experiment we calculated the mean face and the Eigenfaces using only 39 persons, the first image of each person. Using this model the mask is fit the image of the person that was left out. Experiment number three uses the face model, that is trained on the first image of each person, to fit the mask to the fifth image of each of the 40 persons, which are illuminated with a spotlight. Again we use all 4 different masks as well as dividing the original image resolution of  $640 \times 480$  pixels in half three times.

In the last experiment on the IMM Face Database we compared our method with the method proposed in [16]. The approach finds significant facial feature points using an Active Shape Model (ASM) first and then a 3D mesh model is fitted to these points. We therefore used the 58 landmarks of the first image of each person in the IMM Face Database to train the ASM [6]. We use the ASM Matlab code provided by [13]. This trained ASM model is then used to retrieve these 58 landmarks in each of the 40 images again. The ASM face model is initialised to fill the cropped image window found by the frontal face detector [21]. After the landmarks are retrieved by ASM, the CANDIDE-3 mask is fitted to these landmarks as described earlier.

	Mean Vertex Difference
<b>IMM Face Database</b>	
(1) Testing on training	2.6 pixels
(2) Leave one out	4.2 pixels
(3) Lighting variety	4.8 pixels
(4) ASM comparison	6.2 pixels (26/40)
<b>Yale Face Database</b>	
Subset 1	6.5 pixels
Subset 2	6.2 pixels (433/456)

**Table 1.** Mean Euclidean vertex point differences in pixel on images of size  $640 \times 480$ , using mask 2 and 10,000 particles. The mean error of the ASM is calculated for only 26 acceptable mask fits and for the subset 2 of the Yale Face Database only 433 mask fits were reasonable. The face model that was used to fit the mask to the Yale Face Database was trained entirely from the IMM Face Database.

We further use the Yale Face Database B and its extension to conduct more experiments. This database contains images of 10 persons under 9 different poses and 64 different illumination conditions; the extended Yale Face Database B contains additional 28 persons under the same pose and lighting conditions. We use both sets. Furthermore the images for each person are divided into subsets according to the angle between the light source and the camera axis. We use subsets 1 which contains 7 images of each person with spot lights at up to  $12^\circ$  and subset 2 which contains 12 images with angles up to  $25^\circ$ . The greater the angle the longer the shadows on the person’s face.

We used the face model that was built from 40 images of 40 persons in the IMM Face Database to fit the mask to all 266 images of the first subset. Again

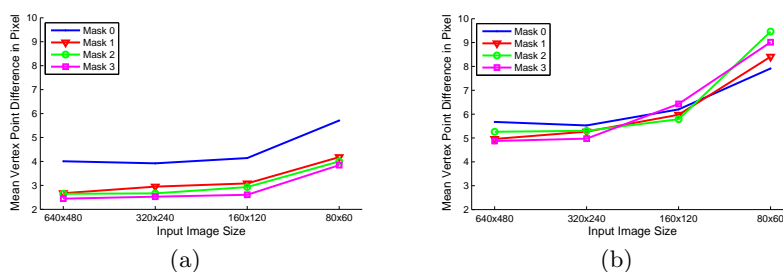


the resolution of the images ( $640 \times 480$  pixels), it is cut in half three times to result in images of size  $320 \times 240$ ,  $160 \times 120$  and  $80 \times 60$  pixels, with corresponding average face sizes of  $250 \times 160$ ,  $125 \times 80$ ,  $60 \times 40$  and  $30 \times 20$  pixels. We also fitted the mask to the second subset, consisting of 456 images, 12 for each person using the original image resolution of  $640 \times 480$  pixels.

As a ground truth we fitted the mask to all 38 individuals of the Yale Face Database B and its extension manually by adjusting the pose and shape parameters. We thereby assume that the pose and shape of each face remained constant across different lighting settings. Again we use the mean Euclidean distance between the manually fitted and the automatically detected mask vertices as a quality measurement.

Our results are summarized in table 1. The error is the mean Euclidean distance in pixels between the manually fitted and the automatically detected mask vertices. Images are  $640 \times 480$  pixels, we use mask 2 and 10,000 particles.

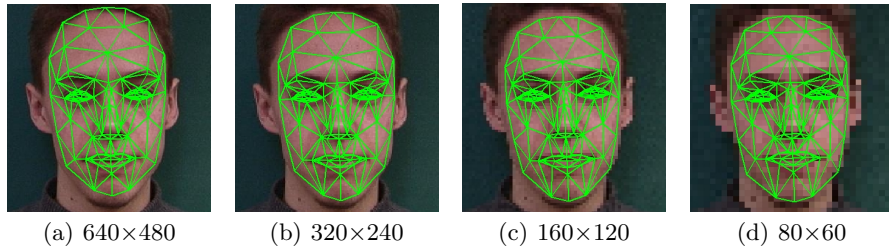
We tested the number of particles versus the processing time needed to achieve best results and found that 10,000 particles is the best trade off between accuracy and speed. The processing time depends on the size of the mask, ie the number of mask triangles. The evaluation of 10,000 particles in Matlab on a Intel Core 2 Quad 2.33GHz PC (using only a single core) takes about 9 s, 20 s, 66 s and 315 s for mask 0, 1, 2 and 3 respectively. The face detection algorithm in Matlab takes 0.40 s and the fast grid-search takes about 0.15 s per image.



**Fig. 5.** Mean vertex point difference for different input resolutions and different mask sizes used for (a) testing on training and (b) testing on different lighting condition.

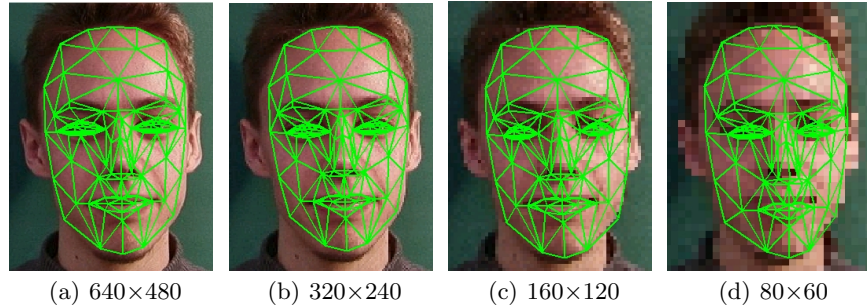
Fig. 5 shows our results for the testing on training experiment and the testing on different lighting condition in more detail. The graph shows the mean vertex point differences averaging over 40 persons for different image resolutions and mask sizes. It shows that the original mask 0 is too coarse and results in the least accurate fitting. Since the mean vertex point difference depends on the image resolution we calculated it with respect to a image resolution of  $640 \times 480$  in order to make the results comparable. Therefore the recovered pose and shape parameters for resolutions  $320 \times 240$  and below are used to project the mask into the image of size  $640 \times 480$ . The graph shows more clearly that the error increases with decreasing image resolution as well as with decreasing mask size. The finer the mask and the higher the resolution of the input image the better the estimation of the mask fit.

We use seven different shape parameters from which some are estimated better than others. Shape parameters controlling positions, namely ‘*eyes vertical position*’, ‘*eye separation distance*’, ‘*nose vertical position*’, ‘*mouth vertical position*’, are estimated more precisely than shape parameters that only apply to a few number of triangles, like ‘*eyebrows vertical position*’, ‘*eyes width*’, ‘*mouth width*’, which are more difficult to estimate.



**Fig. 6.** Result of the testing on training for different image resolutions.

Figure 6 shows the result of testing on training experiment for one person of the IMM Face Database using different image resolutions. The mean vertex point differences are 1.86, 0.54, 0.77 and 0.31 for image resolutions of  $640 \times 480$ ,  $320 \times 240$ ,  $160 \times 120$  and  $80 \times 60$  pixels respectively. Note that the face model is only trained with images of resolution  $640 \times 480$  but by using different mask sizes it can also be applied to image resolutions different from the training resolution.



**Fig. 7.** Result of the proposed automatic fitting for different image resolutions

In comparison Fig. 5(b) shows the results for the testing on different lighting condition, using the fifth image of each person. Again masks 1, 2 and 3 achieve best results for images resolutions of  $640 \times 480$  and  $320 \times 240$  pixels. But modelling the lighting condition is more difficult in low resolution, so the performance decreases for images of size  $160 \times 120$  and below. On average the mean vertex difference increases by about two pixel when fitting the mask to images that exhibit spot lights compared to images with diffuse lighting only. Fig. 7 shows the result for one person of the IMM Face Database for all 4 different resolutions.

When comparing our results with the ASM, we found that the ASM is very sensitive to the initialisation and the parameters used. We found that it was

only able to find the facial feature points of 26 persons correctly, in 14 out of 40 images the ASM drifted completely off the face region using the original resolution of  $640 \times 480$ . After fitting the face mask the mean vertex distance for these 26 persons amounted to 6.2 pixels compared to 2.7 pixels achieved by our approach. As it must be trained on the original image resolution, the ASM was unable to find the facial feature points in resolutions that differ from the training resolution, unlike our approach.

The generalisation of our approach is shown in experiment number 2 and the experiments on the Yale Face Database. The mean vertex point difference for the leave one out experiment is 4.2 pixels as shown in table `reftab:results`. This is only an increase of 1.6 pixels compared to testing on training. The results on the subset 1 of the Yale Face Database show another increase of two pixels. But non of the 38 persons in the Yale Face Database B are included in the face model and all images are recorded under different non-diffuse lighting conditions.

When using the second subset of the Yale Face Database, consisting of 456 images, 12 for each person, our approach was able to correctly fit the mask to 433 of these images, using the original resolution and mask size 2. The mean vertex difference amounts to 6.2 pixels, compared to 6.5 pixels for full resolution images of the first subset. Results will decrease with decreasing resolution and more extreme lighting conditions.

## 4 Conclusion

We proposed a new method for automatic fitting of a deformable face mask to a previous unseen face. By subdividing a coarse mask mesh to produce different mask sizes we are able to fit this mask to different input image resolutions. The face model is built with the highest image resolution and can then be applied to any smaller resolution unlike ASM. Using different mask sizes we are able to fit a deformable mask to faces with average face sizes of down to  $30 \times 20$ .

Traditional approaches usually find facial features first before fitting a face mask to these points whereas our method combines these two steps. The fitting of a deformable face mask implies the detection of facial feature points. By incorporating the work of [3] the proposed method can handle different lighting conditions as shown in experiments on the IMM and the extended Yale Face Database. The approach is shown to generalise well even when testing on a completely different data set to the training data. Future work involves extending the approach to handle facial expressions and partial occlusions of the face.

## References

1. J. Ahlberg. CANDIDE-3 - an updated parameterized face. Technical Report LiTH-ISY-R-2326, Dept. of Elec. Eng., Linkoping University, Sweden, 2001.
2. S. Ayala Raggi, L. Altamirano Robles, and J. Cruz Enriquez. Towards an illumination-based 3D active appearance model for fast face alignment. In *CIARP*, pages 568–575, 2008.
3. R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:218–233, 2003.

4. V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, 2003.
5. D. Cai. Codes and datasets for subspace learning. <http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html>, As retrieved in October 2007.
6. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comp. Vis. Image Underst.*, 61(1):38–59, 1995.
7. G. Dedeoglu, S. Baker, and T. Kanade. Resolution-aware fitting of active appearance models to low-resolution images. In *European Conference on Computer Vision*, volume II, pages 83 – 97. Springer-Verlag, May 2006.
8. J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005.
9. F. Dornaika and J. Ahlberg. Fitting 3D face models for tracking and Active Appearance Model training. *Image and Vision Computing*, 24(9):1010–1024, 2006.
10. A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
11. G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Face & Gesture Recognition*, page 300. IEEE, 1998.
12. A. Georgiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.
13. G. Hamarneh. Multi-resolution active shape models. <http://www.cs.sfu.ca/~hamarneh/software/asm/index.html>, As retrieved in February 2008.
14. D. Kalinkina, A. Gagalowicz, and R. Roussel. 3d reconstruction of a human face from images using morphological adaptation. In *MIRAGE 2007*, LNCS, INRIA Rocquencourt, France, march 2007. Springer Verlag.
15. A. Kuhl, T. Tan, and S. Venkatesh. Model-based combined tracking and resolution enhancement. In *Intl. Conf. on Pattern Recognition*. IEEE, December 2008.
16. L. Lu, Z. Zhang, H.-Y. Shum, Z. Liu, and H. Chen. Model- and exemplar-based robust head pose tracking under occlusion and varying expression. In *IEEE Workshop on Models versus Exemplars in Computer Vision*, 2001.
17. M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann. The IMM face database - an annotated dataset of 240 face images. Technical report, Technical University of Denmark, DTU, May 2004.
18. H. Tang and T. S. Huang. MPEG4 performance-driven avatar via robust facial motion tracking. In *IEEE International Conference on Image Processing*, 2008.
19. J. Tu, Z. Zhang, Z. Zeng, and T. Huang. Face localization via hierarchical condensation with fisher boosting feature selection. In *CVPR*, pages II: 719–724, 2004.
20. M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.
21. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
22. J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2D+3D active appearance models. In *IEEE Comp Vis and Pattern Recog*, pages 535–542, 2004.
23. L. Xin, Q. Wang, J. Tao, X. Tang, T. Tan, , and H. Shum. Automatic 3D face modeling from video. In *International Conference on Computer Vision*, volume 2, pages 1193–1199. IEEE, October 2005.
24. S. Yan, M. Li, H. Zhang, and Q. Cheng. Ranking prior likelihood distributions for bayesian shape localization framework. In *Intl Conf on Comp Vis*. IEEE, 2003.
25. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
26. D. Zorin and P. Schroder. Subdivision for modeling and animation. In *Subdivision for modeling and animation*, number 36 in Computer Graphics and Interactive Techniques. ACM Siggraph, July 2000.