International Journal of Computer Vision 75(2), 297–309, 2007

DOI: 10.1007/s11263-006-0034-8

A Unified Gradient-Based Approach for Combining ASM into AAM

JAEWON SUNG
Department of Computer Science and Engineering, POSTECH, San 31, Hyoja-Dong, Nam-Gu, Pohang, 790-784, Korea
jwsung@postech.ac.kr

TAKEO KANADE
Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue Pittsburgh, PA 15213
tk@cs.cmu.edu

DAIJIN KIM*
Department of Computer Science and Engineering, POSTECH, San 31, Hyoja-Dong, Nam-Gu, Pohang, 790-784, Korea
dkim@postech.ac.kr

Received May 4, 2006; Accepted December 29, 2006

First online version published in January, 2007

Abstract. Active Appearance Model (AAM) framework is a very useful method that can fit the shape and appearance model to the input image for various image analysis and synthesis problems. However, since the goal of the AAM fitting algorithm is to minimize the residual error between the model appearance and the input image, it often fails to accurately converge to the landmark points of the input image. To alleviate this weakness, we have combined Active Shape Models (ASM) into AAMs, in which ASMs try to find correct landmark points using the local profile model. Since the original objective function of the ASM search is not appropriate for combining these methods, we derive a gradient based iterative method by modifying the objective function of the ASM search. Then, we propose a new fitting method that combines the objective functions of both ASM and AAM into a single objective function in a gradient based optimization framework. Experimental results show that the proposed fitting method reduces the average fitting error when compared with existing fitting methods such as ASM, AAM, and Texture Constrained-ASM (TC-ASM) and improves the performance of facial expression recognition significantly.

Keywords: AAM, ASM, combining AAM into ASM, gradient-based optimization, facial expression recognition

1. Introduction

Since ASM (Cootes et al., 1995) and AAM (Cootes et al., 2001; Matthews and Baker, 2004) were introduced, many researchers have focused on these methods to solve many image interpretation problems, especially for facial and medical images (Dornaika and Ahlberg, 2003; Lanitis et al., 1997; Kuilenburg et al., 2005; Thodberg and Rosholm, 2001; Ginneken et al., 2002). ASM and AAM have some similarities. They use the same underlying statistical model of the shape of target objects, represent the shape by a set of landmark points and learn the ranges of shape variation from training images. However, the two methods have several differences as well (Cootes et al., 1999):

1. ASM only models the image texture in the neighboring region of each landmark point, whereas AAM uses the appearance of the whole image region.
2. ASM finds the best matching points by searching the neighboring region of the current shape positions, whereas AAM compares its current model appearance

*Corresponding author.

to the appearance sampled at the current shape positions in the image.

3. ASM seeks to minimize the distance between model points and the identified match points, whereas AAM minimizes the difference between the synthesized model appearance and the target image.

Cootes et al. (1999) found that ASM is faster and has a broader search range than AAM, whereas AAM gives a better match to the texture. However, AAM is sensitive to the illumination condition, especially when the lighting condition in the test images is significantly different from that in the training images. It often fails to locate boundary points correctly if the texture of the object around the boundary area is similar to the background image (Yan et al., 2002). Until now, ASM and AAM have been treated as two independent methods in most cases even though they share some basic concepts such as the same linear shape model and the same linear appearance model (here, the term *appearance* is used with a somewhat broad meaning; it can represent the whole texture or the local texture).

As pointed out by Cootes and Taylor (2001) and Scott et al. (2003), the existing intensity-based AAM has some drawbacks: It is sensitive to changes in lighting conditions and it fails to discriminate noisy flat textured area and real structure, and thus may not lead to accurate fitting in AAM search. To alleviate this problem, Stegmann and Larsen (2002) proposed a few methods: Augmenting the appearance model using extra *feature band* that includes color and gradient channels. Scott et al. (2003) proposed to use some nonlinear transforms such as cornerness, edgeness, and gradient directions (Scott et al., 2003) for the *feature band*. Experimental results showed that the nonlinear descriptions of local structure for the texture model improved the fitting accuracy of the AAMs. Ginneken et al. (2006) pointed out the problem of AAM formulation where only the object's interior is included into the appearance model, which means that the cost function can have minimum value when the model is completely inside the actual object. To avoid this, they simply concatenated the appearance vector with the texture scanned along *whiskers* that are outgoing normal direction at each landmark points.

Another approach to improve the fitting performance is to combine the ideas of the AAM and ASM. Yan et al. (2002) proposed the TC-ASM that inherited the ASM's local appearance model because of its robustness to varying light conditions. They also borrowed the AAM's global texture, to act as a constraint over the shape and providing an optimization criterion for determining the shape parameters. In TC-ASM, the conditional distribution of a shape parameter given its associated texture parameter was modeled as a Gaussian distribution. There was a linear mapping $\vec{s}_t = \mathbf{R}\vec{t}$ between the texture $\vec{t}$ and its corresponding shape $\vec{s}_t$, where $\mathbf{R}$ is a projection matrix that can be pre-computed from the training pairs $\{(\vec{s}_i, \vec{t}_i)\}$. The search stage computes the next shape parameters by interpolating the shape from a traditional ASM search and the texture-constrained shape. Using the texture constrained shape enabled the search method to escape from the local minima of the ASM search, resulting in improved fitting results.

In this paper, we propose a new fitting method that integrates AAM and ASM in a unified gradient-based optimization framework. The goal of ASM is to find shape parameters so that the profile at each model point is similar to the pre-learned profile, while keeping the shape parameters within the learned range in the parameter space. The goal of AAM is to find the shape and appearance model parameters such that the model instance is most similar to the input image. The model instance obtained from the current model parameter is as similar as the input image.

One simple and direct combination of ASM and AAM is to alternate between the two. In this case, the parameters may not converge to a stable solution because they use different optimization goals and techniques. To guarantee a stable and precise convergence, we changed the profile search step of the ASM to a gradient-based search like the AAM search method and combined the error terms of the AAM and ASM into a single objective function in a gradient-based optimization framework. The gradient-based ASM search method can be seen as a simplified version of the Active Contour Model (ACM) (Kass et al., 1998), because the active contour model uses all boundary points while the gradient-based ASM uses only the specified model points.

Figure 1 shows how the proposed fitting method[1] works. The proposed AAM + ASM algorithm

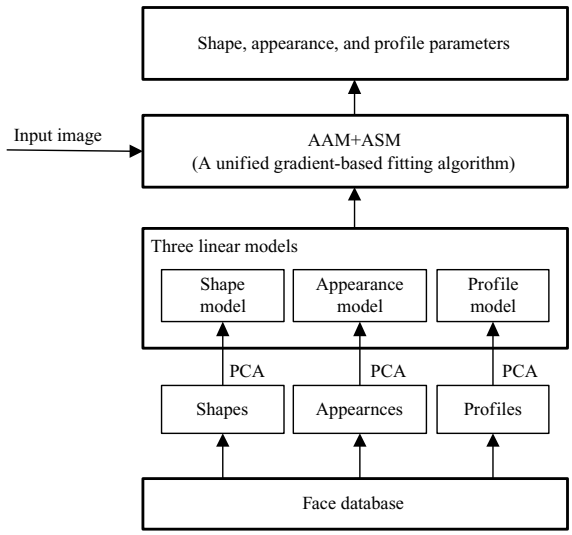

Figure 1. The proposed fitting method.

pre-computes the linear shape, appearance, and profile model independently from a set of the landmark training images and uses these models simultaneously to find the optimal model parameters for an input image. By integrating the AAM and ASM error terms and optimizing them simultaneously, we could obtain more accurate solutions than using only the AAM error term. This improvement is due to the fact that the ASM error term enforces to move the shape points to nearby edge-like points. If we only take the AAM error term, the AAM fitting often fails to converge to the ground truth points because there are no distinctive texture patterns in the cheek area of the face. One thing to note here is that the purpose of the proposed algorithm is not obtaining an illumination robustness fitting but obtaining a more accurate fitting. Experimental results show that the proposed fitting method successfully improves the fitting results in terms of root mean squared (RMS) positional errors when compared to ASM, AAM, and TC-ASM.

This paper is organized as follows. Section 2 briefly reviews the shape and appearance models in ASM and AAM. Section 3 explains the proposed fitting method that incorporates the shape and appearance models of ASM and AAM in a unified framework. Section 4 presents the experimental results and discussion. Finally, Section 5 presents our conclusion.

2. Background

Assume a set of landmarked face images $D = \{I_i, \vec{v}_i\}_{i=1}^N$, where N is the number of images, I_i is the i-th image, and $\vec{v}_i = (x_1, y_1, \ldots, x_v, y_v)^t \in R^{2v\times 1}$ are the coordinates of the landmark points for I_i.

2.1. Shape Model

In ASM and AAM, a shape $\vec{s} = (x_1, y_1, \ldots, x_v, y_v)^t$ is represented as a linear combination of the mean shape $\vec{s}_0$ and n orthonormal bases $\vec{s}_i$ as

$$\vec{s} = \sum_{i=0}^{n} p_i \vec{s}_i, \tag{1}$$

where p_i is the ith shape parameter and $p_0 = 1$. These shape bases are obtained by collecting a set of shape vector $\{\vec{v}_i\}_{i=1}^N$, aligning them by removing the variations due to scaling, rotation, and translation, and applying PCA to the resultant aligned shape vectors. Figure 2 illustrates an example of the shape model, where the five shape bases $\vec{s}_1$ to $\vec{s}_5$ are displayed.

The shape at the j-th landmark point $\vec{s}_j$ $(j = 1, \ldots, v)$ that is synthesized by the shape parameters p_i $(i = 0, \ldots, n)$ and a global similarity transformation $\vec{q} = (q_1, q_2, q_3, q_4)^t$ is given by

$$\begin{aligned}\vec{s}^{\,j} &= \begin{bmatrix} 1+q_1 & q_2 \\ -q_2 & 1+q_1 \end{bmatrix} \left(\sum_{i=0}^{n} p_i \vec{s}_i^{\,j} \right) + \begin{bmatrix} q_3 \\ q_4 \end{bmatrix} \\ &= Q_R \left(\sum_{i=0}^{n} p_i \vec{s}_i^{\,j} \right) + Q_T, \end{aligned} \tag{2}$$

where $\vec{s}_i^{\,j}$ represents a subvector of the i-th shape basis corresponding to the j-th model point.

2.2. Appearance Models

ASM and AAM use different appearance models: the ASM uses a local profile model and the AAM uses a whole appearance model. We introduce these different appearance models in this section.

2.2.1. Local Profile Model. ASM represents the local appearance at each model point by the intensity or gradient profile (Cootes et al., 1995). For each landmark point in the training image, the intensity profile $\vec{I}^{\,j}$ is obtained by sampling the intensity along the direction that is normal to the line that connects two neighboring landmarks of a given landmark point. Then, the gradient profile $\vec{G}^{\,j}$ is obtained, which is a derivative of the intensity profile $\vec{I}^{\,j}$. Figure 3 illustrates an example: (a) landmarks and a normal vector direction at a specific landmark point, (b) the intensity profile, and (c) the gradient profile. In this work, we consider the gradient profile because it is not sensitive to global intensity variations.

The gradient profile $\vec{g}^{\,j}$ of the j-th model point is also represented as a linear combination of a mean gradient

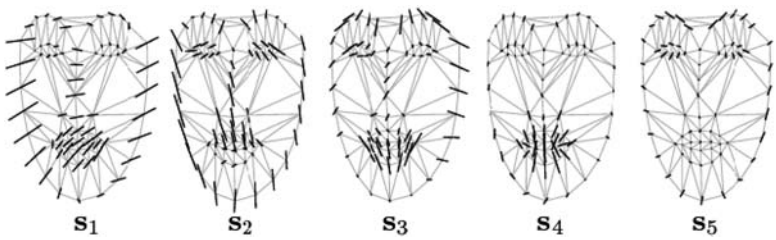

Figure 2. An example of shape bases, $\vec{s}_1$ to $\vec{s}_5$.

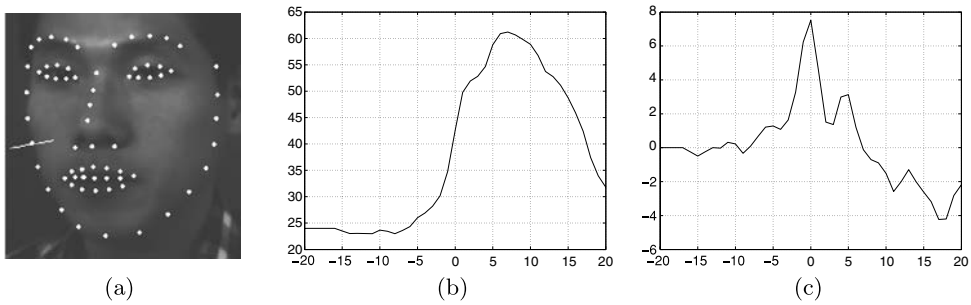

Figure 3. An example of intensity profile (b) and gradient profile (c) along the normal direction indicated in (a).

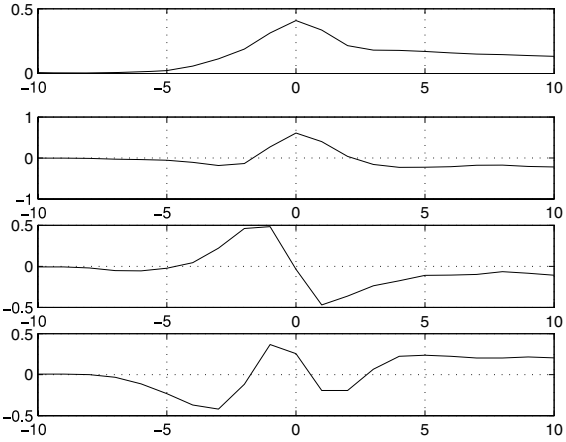

Figure 4. A mean and the first three gradient profile basis vectors.

profile $\vec{g}_0^{\,j}$ and l orthonormal gradient profile basis vectors $\vec{g}_i^{\,j}$ as

$$\vec{g}^{\,j} = \sum_{i=0}^{l} \beta_i \vec{g}_i^{\,j}, \tag{3}$$

where β_i is the i-th gradient profile parameter and $\beta_0 = 1$. The gradient profile basis vectors are obtained by collecting a set of gradient profile vectors $\{\vec{G}_i\}_{i=1}^{N}$, and applying PCA. Figure 4 shows the mean and the first three gradient profile basis vectors. In this figure, the first and second bases appears as a quadrature pair, which implies inaccurate positioning of landmarks on intensity contours. When we collect the gradient profile data, we landmark the position of the feature points manually. Hence, they cannot be aligned precisely. This may produce the quadrant pairs in the basis vectors of the gradient profile. However, this misalignment does not affect the fitting performance much.

2.2.2. Whole Appearance Model. In AAM (Cootes et al., 2001; Matthews and Baker, 2004), the whole appearance is defined on the mean shape $\vec{s}_0$ and the appearance variation is modeled by the linear combination of a mean appearance A_0 and m orthonormal appearance basis vectors A_i:

$$A(\vec{x}) = \sum_{i=0}^{m} \alpha_i A_i(\vec{x}), \tag{4}$$

where α_i is the i-th appearance parameter and $\alpha_0 = 1$. The appearance basis vectors are computed by applying PCA to the shape normalized appearance images that are warped to a mean shape $\vec{s}_0$ from landmarked training images using piece-wise affine warping (Matthews and Baker, 2004). Figure 5 illustrates an example of the appearance model, where the five appearance basis vectors added to the mean appearance are shown.

Although Cootes et al. (2001) proposed the *combined AAM*, in which a third linear model is used to represent the variations of the shape and appearance parameters simultaneously, we have adopted the *independent AAM* scheme, in which independent shape and appearance models are used.

Figure 5. Examples of the five modes of appearance variation.

3. A Unified Approach

Our goal is to find the model parameters that minimize the residual error of the whole and local appearance model simultaneously in a unified gradient-based framework. This required the definition of an integrated objective function that combines the objective functions of AAM and ASM in an appropriate manner.

3.1. Integrated Objective Function

The objective function of the proposed method consists of three error terms: the error of whole appearance E_{aam}, the error of local appearance E_{asm}, and a regularization error E_{reg}. The last error term is introduced to prevent the shape parameters from deviating too widely. We explain each error term and then introduce the overall objective function that consists of the three error terms.

First, we define the error of the whole appearance model for AAM as

$$E_{aam}(\vec{\alpha}, \vec{p}, \vec{q}) = \frac{1}{N} \sum_{\vec{x} \in \vec{s}_0} \left[\sum_{i=0}^{m} \alpha_i A_i(\vec{x}) - I(W(x; \vec{p}, \vec{q})) \right]^2, \tag{5}$$

where N is the number of the pixels $\vec{x} \in \vec{s}_0$, and $\vec{\alpha}$, $\vec{p}$, and $\vec{q}$ are the appearance, shape, and similarity transformation parameters.

Second, we define the error of local appearance model for ASM as

$$\begin{aligned} E_{asm}(\vec{\beta}, \vec{p}, \vec{q}) &= \frac{K}{v \cdot N_{pf}} \sum_{j=1}^{v} \sum_{z} E_{asm}^{j}(z)^2 \alpha^j \\ &= \frac{K}{v \cdot N_{pf}} \sum_{j=1}^{v} \sum_{z} \left\{ \sum_{i=0}^{l} \beta_i^j \vec{g}_i^j(z) \right. \\ &\quad \left. - \vec{g}(W^j(z; \vec{p}, \vec{q})) \right\}^2 \alpha^j, \end{aligned} \tag{6}$$

where N_{pf} is the length of the gradient profile vector, K is a scaling factor to balance the magnitude of E_{asm} with that of the E_{aam}, β_i^j is the i-th gradient profile model parameter corresponding to the j-th model point, $W^j(z; \vec{p}, \vec{q})$ represents a warping function that transforms a scalar coordinate z of the 1-D gradient profile vector into a 2D image coordinate of the image to be used for reading the image gradient profile $\vec{g}$ at each j-th model point, and α^j is adaptive weight control term that will be explained in Section 3.4. The warping function can be represented as

$$W^j(z; \vec{p}, \vec{q}) = \vec{s}^j(\vec{p}, \vec{q}) + z\vec{n}^j, \tag{7}$$

where $\vec{s}^j(\vec{p}, \vec{q})$ and $\vec{n}^j$ are the j-th model point of the current shape corresponding to current shape parameters $\vec{p}$ and $\vec{q}$, and the normal vector of the j-th model point.

Third, we define a regularization error term E_{reg}, which constrains the range of the shape parameters p_i, as

$$E_{reg}(\vec{p}) = R \cdot \sum_{i=1}^{n} \frac{p_i^2}{\sqrt{\lambda_i}^2}, \tag{8}$$

where λ_i is the eigenvalue corresponding to the i-th shape basis $\vec{s}_i$ and R is a constant that controls the effect of regularization term. If the value of R is set to a large value, the fitting result tends to be close to the mean shape. While the shape parameters are directly limited not to exceed $\pm 3\sqrt{\lambda_i}$ after each iteration in ASM (Cootes et al., 1995) and AAM (Matthews and Baker, 2004), we add E_{reg} into the objective function to obtain similar effect.

By combining (5), (6), and (8), we define an integrated objective function E as

$$E = (1 - \omega)(E_{aam} + E_{reg}) + \omega E_{asm}, \tag{9}$$

where $\omega \in [0, 1]$ determines how significant E_{asm} term will be in the overall objective function E. Thus, the proposed algorithm operates like AAM when $\omega = 0$, and like ASM when $\omega = 1$.

3.2. Derivation of Updating Parameters

The integrated objective function is optimized using the Gauss-Newton gradient descent method. To derive the formula for updating parameters, we need to compute the steepest descent vector of the parameters. After the steepest descent vector of each term is obtained, its corresponding Hessian matrix can be easily computed. In the following, we describe the detailed derivation of the steepest descent vectors of the three error terms. During the derivation, we omit the constants $1/N$ in E_{aam} term and $K/(v \cdot N_{pf})$ in E_{asm}.

3.2.1. Steepest Descent Vector of E_{aam}. The whole appearance error term E_{aam} is almost the same as that of the traditional AAM. The difference is that E_{aam} is divided by the number of pixels to effectively balance the error function with E_{asm}. Various gradient based fitting methods for this type of error function have been proposed (Matthews and Baker, 2004; Baker et al., 2003), which are extended from the Lucas-Kanade image matching method (Kanade and Lucas, 1981).

When we take the traditional Gauss-Newton nonlinear optimization method to minimize the first term E_{aam} of Eq. (5), the increments $\Delta\vec{\alpha}$, $\Delta\vec{p}$ and $\Delta\vec{q}$ are determined

to satisfy the following condition:

$$E_{aam}(\vec{\alpha}+\Delta\vec{\alpha},\vec{p}+\Delta\vec{p},\vec{q}+\Delta\vec{q}) < E_{aam}(\vec{\alpha},\vec{p},\vec{q}), \tag{10}$$

where

$$E_{aam}(\vec{\alpha}+\Delta\vec{\alpha},\vec{p}+\Delta\vec{p},\vec{q}+\Delta\vec{q}) = \sum_{\vec{x}\in\vec{s}_0}\left[\sum_{i=0}^{m}(\alpha_i+\Delta\alpha_i)A_i(\vec{x}) - I(W(\vec{x};\vec{p}+\Delta\vec{p},\vec{q}+\Delta\vec{q}))\right]^2 \tag{11}$$

and

$$E_{aam}(\vec{\alpha},\vec{p},\vec{q}) = \sum_{\vec{x}\in\vec{s}_0}\left[\sum_{i=0}^{m}\alpha_i A_i(\vec{x}) - I(W(x;\vec{\vec{p}},\vec{q}))\right]^2. \tag{12}$$

After obtaining the increment vectors $\Delta\vec{\alpha}$, $\Delta\vec{p}$ and $\Delta\vec{q}$, the additive update formula modifies the appearance parameter $\vec{\alpha}$ and the warping parameters $\vec{p}$ and $\vec{q}$ as

$$\vec{\alpha}\leftarrow\vec{\alpha}+\Delta\vec{\alpha},\ W(\vec{x};\vec{p},\vec{q})\leftarrow W(\vec{x};\vec{p}+\Delta\vec{p},\vec{q}+\Delta\vec{q}). \tag{13}$$

We can simply rewrite Eq. (11) as

$$E_{aam} = \sum_{\vec{x}\in\vec{s}_0}\left\{E_{aam}(\vec{x}) + SD_{aam}(\vec{x})\begin{bmatrix}\Delta\vec{\alpha}\\ \Delta\vec{p}\\ \Delta\vec{q}\end{bmatrix}\right\}^2, \tag{14}$$

where

$$E_{aam}(\vec{x}) = \sum_{i=0}^{m}\alpha_i A_i(\vec{x}) - I(W(\vec{x};\vec{p},\vec{q})), \tag{15}$$

and

$$SD_{aam}(\vec{x}) = \left[A_1(\vec{x}),\ldots,A_m(\vec{x}), -\nabla I^t\left(\frac{\partial W}{\partial\vec{p}},\ \frac{\partial W}{\partial\vec{q}}\right)\right] \tag{16}$$

by applying the first order Taylor series expansion to Eq. (11) and simplifying it. Note that the steepest descent vector SD_{aam} must be re-computed at every iteration because the gradient of the image $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^t$ and the Jacobian of the warping function $(\frac{\partial W}{\partial\vec{p}}, \frac{\partial W}{\partial\vec{q}})$ depend on the warping parameters $\vec{p}$ and $\vec{q}$ that are updated at every iteration, i.e., ∇I is computed at $W(\vec{x};\vec{p},\vec{q})$ and the warping function W is differentiated at current $\vec{p}$ and $\vec{q}$. Therefore, the Hessian matrix is also re-computed in each iteration.

3.2.2. Steepest Descent Vector of E_{reg}. As in the case of E_{aam}, we apply the additive Gauss-Newton update to the E_{reg} of Eq. (8), which can be reformulated as

$$E_{reg} = \|\Lambda^{-1}\left(\vec{p}+\Delta\vec{p}\right)\|^2, \tag{17}$$

where Λ is a square diagonal matrix ($\Lambda_{i,i} = \sqrt{\lambda_i}$, and λ_i is the i-th eigenvalue), and it is minimized with respect to $\Delta\vec{p}$. After obtaining the increment vector $\Delta\vec{p}$, the modified additive update formula modifies the shape parameter $\vec{p}$ as $\vec{p}\leftarrow\vec{p}+\Delta\vec{p}$.

When we apply the Taylor series expansion to Eq. (17), we can rewrite it as

$$E_{reg} = \|\Lambda^{-1}\vec{p}+\Lambda^{-1}\Delta\vec{p}\|^2 = \|E_{reg,\vec{p}} + SD_{reg}\Delta\vec{p}\|^2, \tag{18}$$

where

$$SD_{reg} = \Lambda^{-1},\quad E_{reg,\vec{p}} = \Lambda^{-1}\vec{p}. \tag{19}$$

3.2.3. Steepest Descent Vector of E_{asm}. Similarly, we apply the additive Gauss-Newton gradient descent method to the local profile error E_{asm}, i.e., we want to minimize

$$E_{asm} = \sum_{j=1}^{v}\sum_{z}\left\{\sum_{i=0}^{l}\vec{g}_i^{\,j}(z)\left(\beta_i^j+\Delta\beta_i^j\right) - \vec{g}(W^j(z;\vec{p}+\Delta\vec{p},\vec{q}+\Delta\vec{q}))\right\}^2, \tag{20}$$

with respect to $\Delta\vec{\beta}$, $\Delta\vec{p}$, and $\Delta\vec{q}$. After obtaining the increment vectors $\Delta\vec{\beta}$, $\Delta\vec{p}$ and $\Delta\vec{q}$, the parameters are updated:

$$\begin{aligned}\vec{\beta}^j &\leftarrow \vec{\beta}^j+\Delta\vec{\beta}^j,\\ W^j(z;\vec{p},\vec{q}) &\leftarrow W^j(z;\vec{p}+\Delta\vec{p},\vec{q}+\Delta\vec{q}).\end{aligned} \tag{21}$$

When we apply the Taylor series expansion to Eq. (20) and ignore the second and higher order terms, we can rewrite Eq. (20) as

$$E_{asm} = \sum_{j=1}^{v}\sum_{z}\left\{E_{asm}^j(z) + SD_{asm}^j(z)\begin{bmatrix}\Delta\vec{\beta}\\ \Delta\vec{p}\\ \Delta\vec{q}\end{bmatrix}\right\}^2, \tag{22}$$

where

$$E^j_{asm}(z) = \sum_{i=0}^{l} \vec{g}^j_i(z)\beta^j_i - \vec{g}(W^j(z;\vec{p},\vec{q})) \quad (23)$$

and

$$SD^j_{asm}(z) = \left[\vec{g}^j_1(z), \ldots, \vec{g}^j_l(z), -\nabla\vec{g}(z)\left(\frac{\partial W^j}{\partial \vec{p}}, \frac{\partial W^j}{\partial \vec{q}}\right)\right]. \quad (24)$$

The differentiation of the W^j with respect to the warping parameters $\vec{p}$ and $\vec{q}$ in (24) can be computed as

$$\frac{\partial W^j}{\partial \vec{p}} = \left(\frac{\partial W^j}{\partial z}\right)^t \frac{\partial z}{\partial \vec{p}}, \quad \frac{\partial W^j}{\partial \vec{q}} = \left(\frac{\partial W^j}{\partial z}\right)^t \frac{\partial z}{\partial \vec{q}}, \quad (25)$$

where the first term $\frac{\partial W^j}{\partial z}$ can be computed from Eq. (7), and the second term $\frac{\partial z}{\partial \vec{p}}$ and $\frac{\partial z}{\partial \vec{q}}$ are required to represent z as a function of parameters $\vec{p}$ and $\vec{q}$, respectively. When the j-th model point $\vec{s}^j$ moves to $\hat{\vec{s}}^j$ along the normal vector $\vec{n}^j$, the z coordinate of the $\hat{\vec{s}}^j$ can be computed as

$$z = (\vec{n}^j)^t(\hat{\vec{s}}^j - \vec{s}^j) = (\vec{n}^j)^t\left(Q_R\sum_{i=0}^{n}\vec{s}^j_i p_i + Q_T - \vec{s}^j\right). \quad (26)$$

3.3. Parameter Updates

Since the overall objective function is a summation of multiple objective functions and each objective function consists of a sum of squares, the Hessian matrix of the overall objective function is the sum of the Hessian matrix of each objective function (Xiao et al., 2004) as

$$H_{overall} = (1-\omega)\left\{\sum_{\vec{x}} SD_{aam}(\vec{x})^t SD_{aam}(\vec{x}) + SD^t_{reg}SD_{reg}\right\} + \omega\sum_{j=1}^{v}\sum_{y} SD^j_{asm}(y)^t SD^j_{asm}(y). \quad (27)$$

Similarly, the steepest descent update of the overall objective function is also the sum of the steepest descent updates of the objective functions:

$$SD_{overall} = (1-\omega)\left\{\sum_{\vec{x}} SD_{aam}(\vec{x})^t E_{aam}(\vec{x}) + SD^t_{reg}E_{reg,\vec{p}}\right\} + \omega\sum_{j=1}^{v}\sum_{z} SD^j_{asm}(z)^t E^j_{asm}(z). \quad (28)$$

If we define the overall parameter vector as $\vec{\theta} = [\Delta\vec{\alpha}^t\ \vec{\beta}^t\ \vec{p}^t\ \vec{q}^t]^t$, we can compute the overall increment vector $\Delta\vec{\theta} = [\Delta\vec{\alpha}^t\ \Delta\vec{\beta}^t\ \Delta\vec{p}^t\ \Delta\vec{q}^t]^t$ as

$$\Delta\vec{\theta} = -H^{-1}_{overall}SD_{overall}. \quad (29)$$

Once the overall increment vector $\Delta\vec{\theta}$ is obtained, the appearance parameter α, the gradient profile parameter $\vec{\beta}$, and the warping parameters $\vec{p}$ and $\vec{q}$ are updated as

$$\vec{\alpha} = \vec{\alpha} + \Delta\vec{\alpha},\ \vec{\beta} = \vec{\beta} + \Delta\vec{\beta},\ \vec{p} = \vec{p} + \Delta\vec{p},\ \vec{q} = \vec{q} + \Delta\vec{q}. \quad (30)$$

In (29), the sizes of the three matrices are different because the parameter set of the three individual objective functions E_{aam}, E_{reg}, and E_{asm} are different. We can deal with this by thinking that they are the functions of all the entire parameter set and setting all elements in both the Hessian and the steepest descent parameter updates to zero that do not have corresponding entries in the individual functions.

3.4. Adaptive Weight Control of E_{asm}

First, we consider the effect of convergence on E_{asm}. As mentioned earlier, the local profile model for the ASM has only learned the local profile variations that are near the landmark points of the training data. Thus, the weight on E_{asm} term must be controlled appropriately for a proper model fitting in the following manner. During early iteration, the E_{asm} term should have little influence because the synthesized shape is typically far from the landmark points. In the later iterations, as the synthesized shape becomes closer to the landmark points, the effect of E_{asm} should become stronger.

To reflect this idea, we need a measure to indicate how accurately the model shape is converged to the landmark points, and the E_{aam} term meets this requirement well. The degree of convergence is then represented by a bell-shaped function (Jang et al., 1997) of the E_{aam} term:

$$Bell(a,b,c;E_{aam}) = \frac{1}{1+\left|\frac{\sqrt{E_{aam}}-c}{a}\right|^{2b}}, \quad (31)$$

where a, b, and c parameters determine the width of the bell, the steepness of downhill curve, and the center of the bell, respectively. In this work, we set $c = 0$ to use the right side of the bell-shape. Figure 6 illustrates a typical bell-shape, where the values of a and b, 15 and 5, were determined experimentally.

Second, we consider how well each model point has converged to its landmark point. Although the synthesized shape is converged to the landmark points on

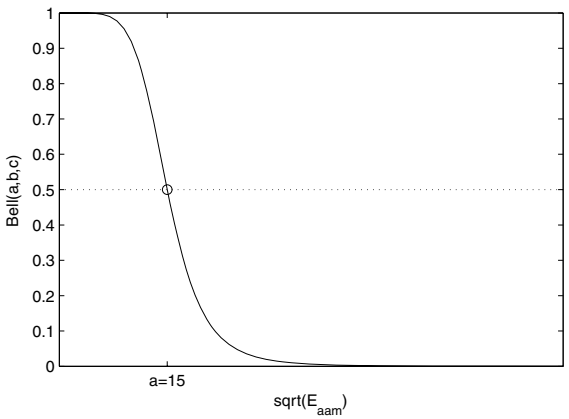

Figure 6. A bell-shaped weight function.

average, some points are close to their landmark points but other points are still far from them. To accommodate this situation, we consider a Normal-like function $\exp(-\frac{E^j_{asm}}{2\sigma^j})$, where $E^j_{asm}(j = 1, \ldots, v)$ is the local profile error at j-th model point, and $\sigma^j > 0$ controls the sensitivity of the normal function, i.e., σ^j determines how much weight will be imposed on the j-th shape point using the current gradient profile error E^j_{asm}. The reason we use different sensitivity control parameter σ^j for each shape point is that the statistics of the gradient profile error are different from point to point. Therefore, we measured the mean value of the gradient profile errors E^j_{asm} at each landmark points from training data and set the σ^j values as the consistently scaled values of the measured statistics.

Considering these two effects, the adaptive weight α^j in (6) is controlled as

$$\alpha^j = Bell(E_{aam}) \cdot \exp\left(-\frac{E^j_{asm}}{2\sigma^j}\right). \tag{32}$$

4. Experiment Results and Discussions

We evaluate the performance of the proposed method in terms of fitting error and facial expression recognition.

4.1. Database

We used our own face database that consists of 80 face images, collected from 20 people with each person having 4 different expressions (neutral, happy, surprised and angry). All 80 images were manually landmarked. The shape, appearance, and gradient profile basis vectors were constructed from the images using the methods explained in Section 2. Figure 7 shows some typical images in the face database.

4.2. Fitting Performance

First, we determined the optimal number of the linear gradient profile basis vectors. For this, we built a linear gradient model using 40 images of 10 randomly selected people, fitted the generated model to them, and measured the average fitting error of 70 landmark points, where the fitting error was defined by the distance between a landmark point and its converged vertex.

Figure 8 shows the average fitting error, where $*$ denotes the mean value of average fitting error when AAM was used, $\circ$ denotes the mean value of the average fitting error at each different number of linear gradient profile basis vectors when gradient-based ASM was used, and the bar denotes the standard deviation of the average fitting error in both methods. This figure shows that (1) the average fitting error of gradient-based ASM is smaller than that of AAM, (2) the optimal number of the linear gradient profile basis vectors is 7, and (3) the minimum average fitting error corresponds to approximately 0.5 pixel. Thus, the number

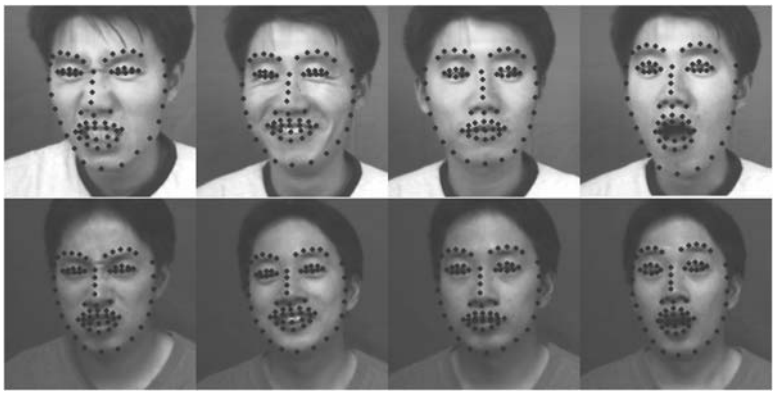

Figure 7. A set of examples face images.

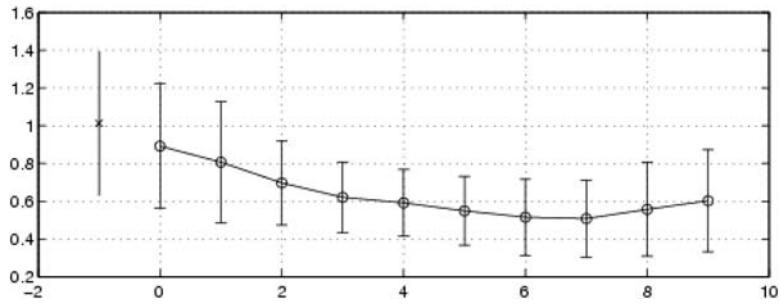

Figure 8. Mean and standard deviation of average fitting errors.

of linear gradient profile basis vectors was 7 in our experiments.

Second, we investigated the effect of the E_{asm} term on the fitting performance. By setting the value of ω to 0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, and 1.0. The scale factor K was set to 10,000 to make the magnitude of the E_{asm} term similar to that of the E_{aam} term (the order of E_{aam} was 10 and the order of E_{asm} was 10^{-3}). For the 40 training images used in the first experiment, the optimal similarity transform parameters were computed from the landmark points, and then the position of the initial shape was moved 3 pixels in a random direction. The initial shape parameters are set to zero.

Figure 9 shows the average fitting error at each ω, where the case of $\omega = 0$ corresponds to the AAM and the case of $\omega = 1.0$ corresponds to the gradient-based ASM. This figure shows that (1) the average fitting error of the AAM could be minimized further by choosing an optimal value of the ω that incorporates the effect of gradient-based ASM and (2) the smallest average fitting error was achieved when ω was set to about 0.5.

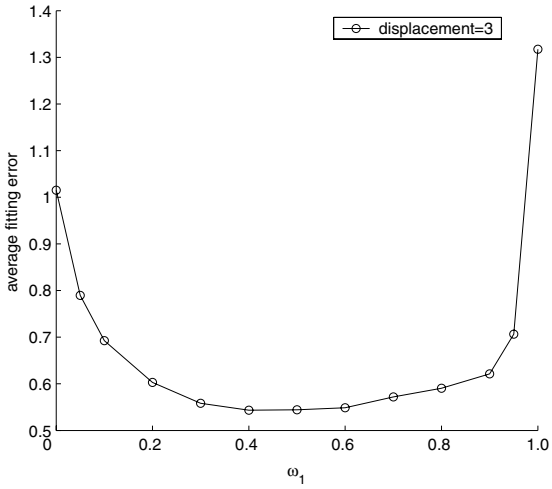

Figure 9. The effect of ω on the average fitting error.

The amount of the decrease in the average fitting error in this experiment is about 0.45 pixel per each model point. Although the improvement seems to be very small, note that the improvement of 0.45 pixel per model point was difficult to achieve because the AAMs worked reasonably under the same conditions in this experiment. The average fitting error of the AAM was about 1.0, which means that every point converged to the ground truth landmark points reasonably. The improvement of 0.45 pixel per model point must be understood as follows: When comparing the average fitting error of the AAM + ASM algorithm to that of the AAM algorithm, the AAM + ASM algorithm reduced the average fitting error to 55%.

In addition, the amount of 0.45 pixel is the average value. Thus, some model points move a lot while other points do not move when the fitting result is compared to that of the AAM algorithm. Usually, the moving points belong to facial features such as mouth, and eye brows. In case of facial expression recognition, extracting the exact location of such points are important because the facial expression can be changed by their small movement. The effect of the accurate fitting result is shown in the facial expression recognition experiment.

Figure 10 illustrates a typical example of fitted results when the AAM and the AAM + ASM ($\omega = 0.5$) were

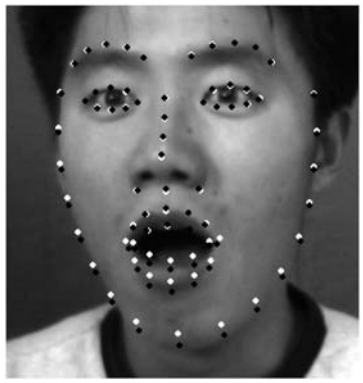

Figure 10. A typical example of fitted results.

used. In this figure, the white and black dots correspond to the fitted results of AAM and AAM + ASM, respectively. The AAM + ASM converged more accurately to the landmark points than the AAM, particularly near the mouth and chin.

Third, we investigated the effect of the a parameter of the Bell-shaped weight control function, where it determines the width of the Bell-shape function. We measured the average fitting error at different values of $a = 5, 10, 15$ and 20. For each value, we performed 8 trials of this experiment using different initial positions, where the displacement is 3 from 8 directions. Figure 11 shows the change of average fitting error with respect to the iteration numbers, where each curve is the mean of 8 trials. This figure shows that (1) if the parameter a is too small (i.e., $a = 5$), the E_{asm} term is not always effective over the range of E_{aam}. This results in only the AAM error term being used, (2) if the parameter a is too large (i.e., $a = 20$), the E_{asm} term is always effective over the range of E_{aam}. This causes the fitting to stick an incorrect local minima, (3) the minimum average fitting error is obtained when a is 15.

Fourth, we compared the fitting performance of AAM + ASM to existing methods such as the traditional AAM and TC-ASM, another approach combining AAM and ASM. We set $\omega = 0$ for AAM and $\omega = 0.5$ for AAM + ASM. We built three linear models (appearance, shape, and profile) using the 40 training images that were used in previous experiments and measured the fitting performance using the remaining 40 test images. For one test image, we tried 40 different initial positions, as shown in Fig. 12, where they correspond to 5 different distances (3, 5, 7, 9, and 11) and 8 different directions. The initialization was done as follows: the shape and appearance parameters were set to zero, and the scale and position parameters were computed from the landmark points.

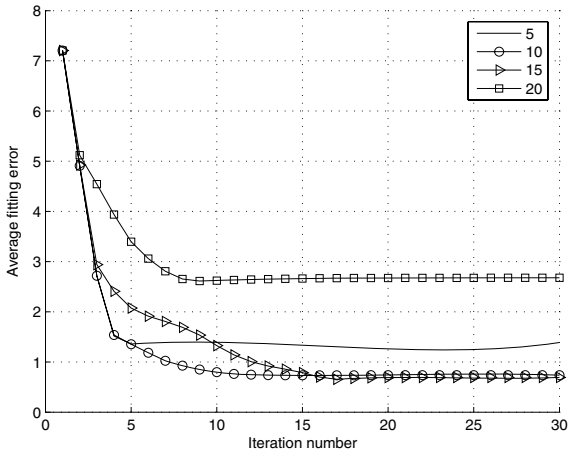

Figure 11. Change of average fitting errors for different a values.

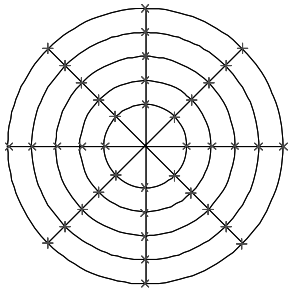

Figure 12. The configuration of 40 initial displacements.

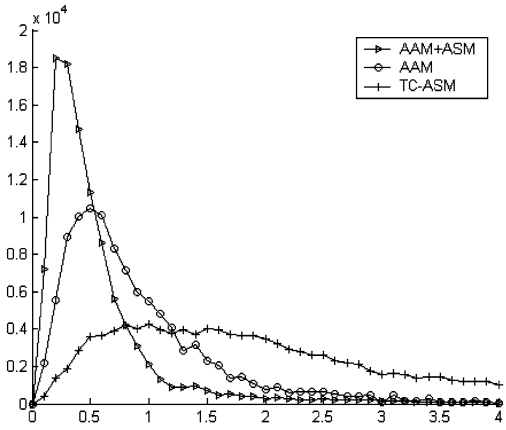

Figure 13. A histogram of fitting errors.

Figure 13 shows a histogram of fitting error for 112,000 cases: 40 images × 40 initial positions × 70 vertices. It shows that the AAM + ASM produced the smallest mean and standard deviation of fitting error.

Table 1 shows the mean value of the fitting error of three different methods, which shows that the AAM + ASM method has the smallest mean value of fitting error.

Figure 14 shows the convergence rates with respect to initial displacements of three different methods, where each plot has the threshold value of 1.5, 2.0, and 2.5, respectively. Here, we assume that the fitting result converges when the average fitting error is less than the given threshold value. In this work, the convergence rate is defined by the rate of the number of converged cases over the number of the all the trials. This figure shows that (1) the convergence rate increases as the threshold value increases for all methods, (2) the convergence rate decreases as the initial displacement increases in the AAM + ASM and the AAM methods, (3) the convergence rate is almost constant as initial displacement increases in

Table 1. Mean value of the fitting error.

	AAM + ASM	AAM	TC-ASM
Mean	0.5	0.8	1.7

Table 2. Comparison of the number of iterations and computation time.

	AAM		AAM + ASM	
Initial displacement	Avg. number of iterations	Avg. computation time (msec)	Avg. number of iterations	Avg. computation time (msec)
3	8.4	879	18.8	4,787
5	9.9	1,033	20.4	4,964
7	11.2	1,162	21.7	5,084
9	12.1	1,250	22.7	5,224

*All algorithm was implemented with Matlab.

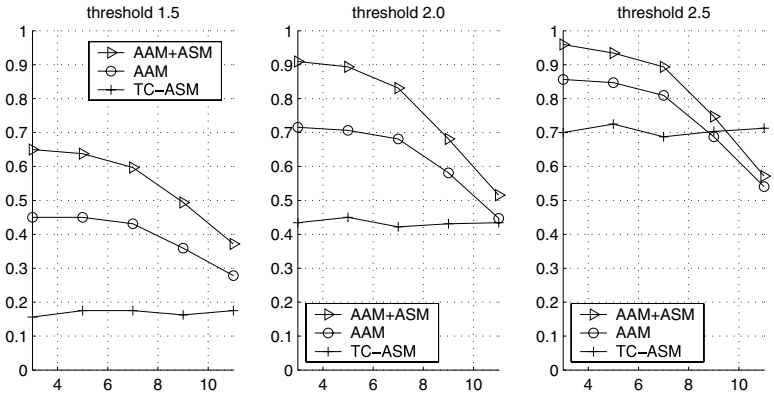

Figure 14. Convergence rate of the three methods.

the TC-ASM method because this method employs a search-based ASM, and (4) the convergence rate of the AAM + ASM is the highest among three methods in almost all cases.

Last, we compared the computation time of the two fitting methods: the AAM and the AAM + ASM in terms of the average number of iterations and the average computation time. For this, we define the convergence of the fitting as follows. When the change of the average Euclidian distance of 70 shape points during two successive iterations is smaller than a certain threshold value of 0.02, it is assumed that the fitting is reached at the convergence. According to a variety of experiments, we obtained that the average computation time per iteration of the AAM and the AAM + ASM method was 104 and 433 milliseconds, respectively. The AAM + ASM algorithm was slower than the AAM method because it had more parameters than the AAM method; both algorithms must update the Hessian and steepest descent vectors whose size was dependent on the number of model parameters, which took the most time of the overall computation.

Table 2 shows the average number of iterations and the average computation time of two fitting methods with respect to the initial displacements. This table shows that (1) the average number of iterations of the AAM method increased as the initial displacement increased, (2) the difference of the average number of iterations between two fitting methods was about 10.5 iterations, which corresponded to about 4 seconds of computation time on Pentium 2.0 GHz CPU, and (3) the AAM + ASM method required more computation time than the AAM by about 4 times.

4.3. *Facial Expression Recognition*

In order to validate the usefulness of the AAM + ASM method, we applied the fitted results to facial expression recognition, where the four facial expressions were neutral, happy, surprised, and angry.

This experiment was conducted as follows. First, we built three linear models (shape, appearance, and gradient profile) from 60 training face images. Second, we computed the shape and appearance parameters of the training images and used them as references for facial expression recognition. Third, we fitted the remaining 20 test face images and obtained the shape and appearance parameters of the test face images. Finally, we determined the facial expression of the test face images based on the distance between the shape and/or appearance parameters of the test face images and the 60 referential shape and/or appearance parameters.

Table 3. Comparison of facial expression recognition rates.

		Features		
Data	Algorithms	Shape	Appearance	Shape + Appearance
Train	AAM	74.6 ± 5.8	100.0 ± 0.0	100.0 ± 0.0
	AAM + ASM	91.7 ± 2.6	100.0 ± 0.0	100.0 ± 0.0
Test	AAM	66.3 ± 8.9	72.5 ± 9.0	82.5 ± 5.6
	AAM + ASM	75.0 ± 6.1	67.5 ± 7.5	90.0 ± 3.5

In this work, we used the simple nearest neighbor approach for facial expression classification. In order to avoid data tweak problem, we used the 4-fold cross validation technique. The recognition performance was evaluated by using shape, appearance, and shape + appearance parameters by changing the number of parameters.

Table 3 shows the mean and standard deviation of facial expression recognition rates for the training and test data. This table shows that (1) the facial expression recognition rate using the test data degrades little from that using the test data, (2) the facial expression recognition rate using the AAM + ASM usually outperforms that of using the AAM, (3) the shape + appearance feature is the best one, and (4) the AAM + ASM using the shape + appearance feature is the best solution for facial expression recognition, producing around 90% recognition rate.

Specifically, the superior fitting performance of AAM + ASM is evident in the classification rates obtained by using shape features, where the classification rate of AAM + ASM is higher than that of AAM by about 17% for training data and 9% for test data. For appearance features, the performance of AAM + ASM is 5% lower than that of AAM for test data. The lower classification rate is somewhat expected because we sacrificed the appearance reconstruction ability to add the profile fitting ability, so that the AAM + ASM tries to fit to an exact shape rather than to reconstruct the appearance. For shape + appearance features, the classification rate of AAM + ASM for test data increased up to about 90%, comparable to that of the ground truth, whereas that of AAM is about 83%.

5. Conclusion

In this paper, we have proposed a unified gradient based framework that combines ASM into AAM and also proposed an adaptive weight control strategy that improved the stability of convergence. Originally, AAM used the whole appearance model and a gradient based approach for model fitting, while ASM used a local profile model and a search based approach for model fitting. Since these properties were not appropriate for combination, we introduced the gradient based approach for ASM.

Basically, AAM + ASM method worked similarly to AAM method and it had an additive property that guaranteed more precise convergence to the landmark points by reducing the fitting error due to the incorporated profile error term.

AAM + ASM was similar to TC-ASM from the viewpoint of using both the whole appearance and local profile. While the TC-ASM used the whole appearance to estimate the texture-constrained shape, and its next estimated shape was obtained by interpolating the texture-constrained shape and the shape estimated by a traditional ASM search, AAM + ASM used the whole appearance and the profile information simultaneously within a gradient-based optimization framework.

Extensive experimental results validated the usefulness of the AAM + ASM method because it reduced the fitting error and improved the facial expression recognition significantly.

Currently, we have to manually determine a set of parameters such as ω and σ^j to obtain the best performance, and their optimal values may be different from one set of data and another. In the future, we will try to develop more generally applicable methods by designing parameter-free weight control mechanisms. Furthermore, it may be possible to implement the proposed algorithm more efficiently by incorporating the gradient-based ASM search in the inverse compositional AAM fitting method.

Acknowledgment

This work was partially supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea. Also, it was partially supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University.

Note

1. Hereafter, we call it the AAM + ASM method.

References

Baker, S., Gross, R., and Matthews, I. 2003. Lucas-Kanade 20 years on: a unifying framework: part 3. CMU-RI-TR-03-05.

Cootes, T., Cooper, D., Taylor, C., and Graham, J. 1995. Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Cootes, T., Edwards, G., and Taylor, C. 1999. Comparing active shape models with active appearance models. In *British Machine Vision Conference*.

Cootes, T., Edwards, G., and Taylor, C. 2001. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Cootes, T. and Taylor, C. 2001. On representing edge structure for model matching. In *Conference on Computer Vision and Pattern Recognition*, 1:1114–1119.

Dornaika, F. and Ahlberg, J. 2003. Face model adaptation for tracking and active appearance model. In *British Machine Vision Conference*.

Ginneken, B., Frangi, A., Staal, J., Romeny, B., and Viergeber, M. 2002. Active shape model segmentation with optimal features. *IEEE Trans. on Medical Imaging*, 21(8):924–933.

Ginneken, B.V., Stegmann, M.B., and Loog, M. 2006. Segmentation of anatomical structures in chest radiographs using supersvised methods: A comparative study on a public database. *Medical Image Analysis*, 10(1):19–40.

Jang, J., Sun, C., and Mizutani, E. 1997. *Neuro-Fuzzy and Soft Computing*, Prentice Hall.

Kanade, T. and Lucas, B. 1981. An ierative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, 1:674–679.

Kass, M., Witkin, A., and Terzopoulos, D. 1998. Snakes: Active ontour models. *Interntaional Journal of Computer Vision*, (4):321–331.

Kuilenburg, H., Wiering, M., and Uyl, M. 2005. A model based method for automaic facial expression recognition. In *European Conference on Maching Learning*.

Lanitis, A., Taylor, C., and Cootes, T. 1997. Automatic interpretation and aoding of face images using flexible models. *IEEE Trans. on Pattern Analysis and Maching Intelligence*, 19(7):743–756.

Matthews, I. and Baker, S. 2004. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164.

Scott, I., Cootes, T., and Taylor, C. 2003. Improving appearance odel matching using local image structure. In *Conference on Information Processing in Medical Imaging*, 2732:258–269.

Stegmann, M.B. and Larsen, R. 2002. Multi-band modelling of appearance. In *International Workshop on Generative Model-Based Vision*.

Thodberg, H.H. and Rosholm, A. 2001. Application of the active shape model in a commercial medial device for bone densitometry. In *British Machine Vision Conference*.

Xiao, J., Baker, S., Matthews, I., and Kanade, T. 2004. Real-time combined 2D + 3D active appearance models. In *International Conference on Computer Vision and Pattern Recognition*.

Yan, S., Liu, C., Li, S., Zhang, H., Shum, H., and Cheng, Q. 2002. Texture-constrained active shape models. In *European Conference on Computer Vision*.