Subspace Analysis and Optimization for AAM Based Face Alignment

Ming Zhao Chun Chen
College of Computer Science
Zhejiang University
Hangzhou, 310027, P.R.China
zhaoming1999@zju.edu.cn

Stan Z. Li
Microsoft Research Asia
Beijing Sigma Center
Beijing, 100080, P.R.China
szli@microsoft.com

Jiajun Bu
College of Computer Science
Zhejiang University
Hangzhou, 310027, P.R.China
bjj@zju.edu.cn

Abstract

[1] *Active Appearance Models (AAM) is very powerful for extracting objects, e.g. faces, from images. It is composed of two parts: the AAM subspace model and the AAM search. While these two parts are closely correlated, existing efforts treated them separately and had not considered how to optimize them overall. In this paper, an approach is proposed to optimize the subspace model while considering the search procedure. We first perform a subspace error analysis, and then to minimize the AAM error we propose an approach which optimizes the subspace model according to the search procedure. For the subspace error analysis, we decomposed the subspace error into two parts, which are introduced by the subspace model and the search procedure respectively. This decomposition shows that the optimal results of AAM can be achieved only by optimizing both of them jointly rather than separately. Furthermore, based on this error decomposition, we develop a method to find the optimal subspace model according to the search procedure by considering both the two decomposed errors. Experimental results demonstrate that our method can find the optimal AAM subspace model rapidly and improve the performance of AAM significantly.*

1. Introduction

Accurate alignment of faces is very important for extraction of good facial features for success of applications such as face recognition, expression analysis and face animation. Active Shape Models (ASM) [7] and Active Appearance Models (AAM) [3, 8, 6] are two successful methods for face alignment. ASM uses the local appearance model, which represents the local statistics around each landmark to efficiently find the target landmarks. And the solution space is constrained by the properly trained global shape model. AAM combines constraints on both the shape and texture. The result shape is extracted by minimizing the texture reconstruction error. According to the different optimization criteria, ASM performs more accurately in shape localization while AAM gives a better match to image texture. In this paper, we will concentrate on AAM.

AAM is composed of two parts: the AAM subspace model and the AAM search, which are treated separately in its original form. The subspace model is trained without considering the search procedure while the search is performed using this subspace model without considering how it is trained. However, we find that these two parts are closely interrelated and the performance depends on both of them. Unfortunately, this relationship is often neglected by previous works. Most efforts [2, 5, 10, 4] only attempted to improve the search procedure without considering the subspace model. Stegmann *et al* [14] attempted to optimize the subspace model, but he did not consider the search procedure. In this paper, we try to optimize the subspace model according to the search procedure.

Because of neglecting the correlation, the dimensionalities of the subspaces in the AAM subspace model are usually chosen to explain as high as 95%∼98% [3, 6, **?**, 4] of the variations. The underlying assumption is that if the subspace reconstruction errors are small, the overall AAM error will be small too. This underlying assumption was taken for granted by previous works without any justification. However, our analysis of AAM subspace errors shows that this is incorrect. To minimize the AAM overall error, we should not only consider the subspace model, but also the search procedure.

In this paper, we present an analysis of AAM subspace errors, and propose an approach for optimizing the parameterization of the AAM subspace model according to the search procedure. First, we identify that the dimensionalities of the subspaces in AAM subspace model significantly affect the performance of AAM. Then, we decompose the subspace error into one of subspace reconstruction and one

1 This paper is supported by National Natural Science Foundation of China (60203013).

of AAM search. This decomposition shows that the best result can only be achieved by optimizing both of them. Finally, we develop a method to optimize the AAM subspace model based on the subspace error decomposition. In this method, we use a new criterion to select the eigenvectors for the subspaces considering both the subspace model and the search procedure, and propose a practical method to implement it, which is proven to be rapid and accurate. By optimizing the subspaces in AAM, we can get more accurate and faster results. What's more, it can be used to optimize other alignment algorithms, such as ASM and AAA [1].

The rest of the paper is arranged as follows. An overview of AAM is described in Section 2. Section 3 discusses the problem of optimizing the subspaces in AAM. In Section 4, we present the decomposition and analysis of the AAM subspace errors. And section 5 gives the method to find the optimal subspaces. Experimental results are provided in Section 6 before conclusions are drawn in Section 7.

2. AAM Modelling and Search

The AAM algorithm contains two parts: the AAM subspace model and the AAM search. The AAM subspace model is composed of three subspaces: the shape, texture and appearance subspace. The AAM search uses learned prediction matrices to find target appearances in images.

2.1. AAM Subspace Modelling

Assume that the training set is given as $\{(S_0, T_0)\}$ where a shape $S_0 = ((x_1, y_1), \ldots, (x_K, y_K)) \in \mathbb{R}^{2K}$ is a sequence of K points in the 2D image plane, and a texture T_0 is the patch of image pixels enclosed by S_0.

Shapes $\{S_0\}$ are aligned to the tangent shape space $\{S\}$ in a common co-ordinate frame with the Procrustes Analysis [9]. The AAM shape subspace is trained by PCA analysis on the tangent shape space

$$S = \bar{S} + \mathbf{P}_s s \tag{1}$$

where $\mathbf{P}_s$ is the matrix of the k principal orthogonal modes of variation in $\{S\}$. Any shape in the AAM shape subspace, denoted $\mathbb{S}_s \subset \mathbb{R}^k$, is represented as a vector $s = \mathbf{P}_s^T(S - \bar{S})$.

After deforming each training shape S_0 to the mean shape, the corresponding texture T_0 is warped and normalized to T. Then all of them are aligned to the tangent space $\{T\}$ of the mean texture $\bar{T}$ by using an iterative approach as described in [3]. The AAM texture subspace is obtained by PCA analysis

$$T = \bar{T} + \mathbf{P}_t t \tag{2}$$

where $\mathbf{P}_t$ is the matrix consisting of l principal orthogonal modes of variation in $\{T\}$. Any texture in the AAM texture subspace, denoted $\mathbb{S}_t \subset \mathbb{R}^l$, is represented as a vector $t = \mathbf{P}_t^T(T - \bar{T})$

Since there may be correlations between the shape and texture variations, the AAM appearance subspace is built from $\mathbb{S}_s$ and $\mathbb{S}_t$. The appearance of each example is a concatenated vector

$$A = \begin{pmatrix} \Lambda s \\ t \end{pmatrix} \tag{3}$$

where Λ is a diagonal matrix of weights for the shape parameters allowing for the difference in units between the shape and texture variation. Again, the AAM appearance subspace is obtained by PCA analysis

$$A = \mathbf{P}_a a \tag{4}$$

where $\mathbf{P}_a$ is the matrix consisting of m principal orthogonal modes of variation in $\{A\}$. Any appearance in the AAM appearance subspace, denoted $\mathbb{S}_a \subset \mathbb{R}^m$, is represented as a vector $a = \mathbf{P}_a^T A$

2.2. AAM Search

In AAM, the search is guided by minimizing the difference δT between the normalized texture T_{im} in the image patch and the texture T_a reconstructed from the current appearance parameters.

The AAM assumes that parameter displacements of the appearance a, and the shape transformation v (translation, scale and rotation) and the texture normalization u (scale and offset) are linearly correlated to δT. It predicts the displacements as

$$\delta \mathbf{p} = -\mathbf{R}\mathbf{r}(\mathbf{p}), \quad \mathbf{R} = \left(\frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \frac{\partial \mathbf{r}}{\partial \mathbf{p}} \right)^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{p}}^T \tag{5}$$

where $\mathbf{r}(\mathbf{p}) = \delta T$, $\mathbf{p}^T = (a^T | v^T | u^T)$.

3. Subspace Selection Problem for AAM

Typically, the subspaces of shape, texture and appearance in AAM are selected so that each explains a given proportion, α, of the total variance. We call this proportion, α, the (subspace) *explanation proportion*. Since the variance along the i^{th} principal axis is equal to the corresponding eigenvalue, λ_i, this is easily carried out. To retain a given proportion α of the variation, t modes can be chosen satisfying $\sum_{i=1}^{t} \lambda_i \geq \alpha \sum \lambda_i$.

How to estimate the explanation proportion of the subspace can be treated as a model selection problem. The goal of traditional model selection is to select the model which gives the highest generalization performance. If we choose too simple a model, it is likely to incur a high error on both the training data and the test data. On the other hand, if we choose too complex a model, it is likely to be poor due to

over-fitting. There are two basic categories of techniques for model selection: complexity penalization and hold-out testing (e.g. cross-validation and bootstrapping) [11].

However, there is a little difference between model selection and the estimation of explanation proportions in AAM. As for traditional model selection problem, it is usually the case that the more complex the model, the lower training error. However, for estimating the explanation proportion of the subspace for AAM, there are cases that the more complex the subspace, the higher training error. The reason is that this estimation involves a search procedure - AAM search. So, to the estimation of the subspace explanation proportion, we must consider not only AAM subspaces, but also AAM search. Complexity penalization techniques can hardly be used in this situation. Although cross-validation can be used for this purpose, it is time-consuming and almost unpractical, for there are three subspaces to be selected. To avoid this problem, Stegmann *et al* [14] used parallel analysis (PA) as an alternative for cross-validation. However, as parallel analysis only considered the subspaces without considering the search procedure, the improvement is very limited. In this paper, we will propose an efficient and effective method based on the cross-validation analysis for estimating the explanation proportions in AAM using the error decomposition in the next section.

4. Subspace Error Decomposition and Analysis

The overall AAM error is determined by both the AAM subspace model and the AAM search. But how? No answers have been given in the previous works. Moreover, previous works only consider AAM subspace model and AAM search separately. In this section, we endeavor to discover the mechanism. When it is revealed, we are on the right track to finding an optimized solution. We will first decompose the AAM subspace error into the reconstruction error and the search error, and then analyze both of them. Based on this error analysis, we will develop a method to find the optimal subspace model in the next section.

4.1. Subspace Error Decomposition

Let $\mathbb{S}$ denote the full eigen-space, which could be any space of the shape, texture and appearance in AAM. By ranking the eigenvectors w.r.t the eigenvalues and selecting the first principal components(eigenvectors), we form an orthogonal decomposition of $\mathbb{S}$ into two mutually exclusive and complementary subspaces: the principal subspace P-$\mathbb{S}$, containing the principal components(eigenvectors) and its orthogonal complement - the complement subspace C-$\mathbb{S}$ containing the remaining eigenvectors . In AAM, the principal subspace P-$\mathbb{S}$ could be the shape subspace $\mathbb{S}_s$, the texture subspace $\mathbb{S}_t$ and the appearance subspace $\mathbb{S}_a$. According to this decomposition, $\mathbb{S}$ is an orthogonal space. It can be decomposed into two subspaces: $\mathbb{S}$ = P-$\mathbb{S}$ $\oplus$ C-$\mathbb{S}$, which means that $\mathbb{S}$ is the direct sum of P-$\mathbb{S}$ and C-$\mathbb{S}$. The axes of P-$\mathbb{S}$ and C-$\mathbb{S}$ are the same as $\mathbb{S}$.

Let x denote the target object in $\mathbb{S}$, and x' denote the projection of x in P-$\mathbb{S}$. Let y' denote the search result in P-$\mathbb{S}$. The relationship between them is illustrated in Figure 1. Let the coordinate of x be $(x_1, x_2, \ldots, x_n)$ in $\mathbb{S}$, where n is the dimension of $\mathbb{S}$. Let the coordinate of x' be $(x'_1, x'_2, \ldots, x'_t)$, y' be $(y'_1, y'_2, \ldots, y'_t)$ in P-$\mathbb{S}$, where t is the dimension of P-$\mathbb{S}$.

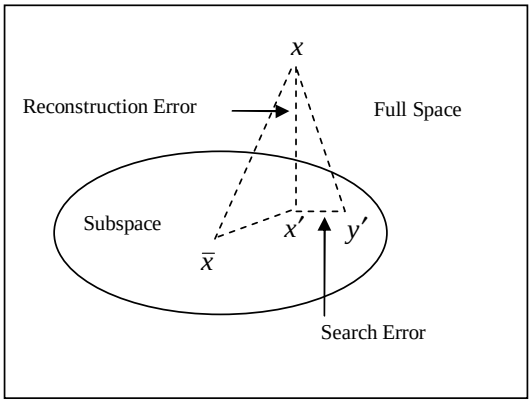


Figure 1. The relationship between the AAM error, the search error and the reconstruction error.

As $\mathbb{S}$ = P-$\mathbb{S}$ $\oplus$ C-$\mathbb{S}$, the axes of P-$\mathbb{S}$ are the first t axes of $\mathbb{S}$. So we can have:

$$x'_i = x_i,\ y'_i = y_i \quad (i = 1 \ldots t) \tag{6}$$

Furthermore, the coordinates of the projection of x' and y' to C-$\mathbb{S}$ are zero, for P-$\mathbb{S}$ is orthogonal to C-TSS. So the coordinates of x' and y' in $\mathbb{S}$ are:

$$(x'_1, x'_2, \ldots, x'_t, 0, \ldots, 0),\ (y'_1, y'_2, \ldots, y'_t, 0, \ldots, 0) \tag{7}$$

The subspace error is defined as the error between the target x and the search result y', i.e.

$$\begin{aligned} ERR_{(x)} &= ||x - y'||^2 = \sum_{i=1}^{n}(x_i - y'_i)^2 \\ &= \sum_{i=1}^{t}(x_i - y'_i)^2 + \sum_{i=t+1}^{n}(x_i - 0)^2 \\ &= \sum_{i=1}^{t}(x'_i - y'_i)^2 + \sum_{i=t+1}^{n}(x_i)^2 \end{aligned}$$

From this formula we can see that the subspace error is composed of two components.

The first component is the square distance between x' and y' as

$$\sum_{i=1}^{t}(x'_i-y'_i)^2=\sum_{i=1}^{t}(x'_i-y'_i)^2+\sum_{i=t+1}^{n}(0-0)^2=||x'-y'||^2 \tag{8}$$

Suppose that if the search procedure is perfect, the search result will be x', for x' is the point with smallest distance in P-$\mathbb{S}$ from x. This error exists just because the search procedure is not perfect. So we call the first component the search error.

The second component is the square distance between x and x' as

$$\sum_{i=t+1}^{n}(x_i)^2=\sum_{i=1}^{t}(x_i-x'_i)^2+\sum_{i=t+1}^{n}(x_i-0)^2=||x-x'||^2 \tag{9}$$

Because x' is the projection of x in P-$\mathbb{S}$, we call this component the reconstruction error.

4.2. Subspace Error Analysis

In AAM, the explanation proportions of the principal subspaces ($\mathbb{S}_s$, $\mathbb{S}_t$, $\mathbb{S}_a$) are usually chosen as high as 95%$\sim$98% [3, 6, **?**, 4]. The underlying assumption is that if the reconstruction error is small, the overall AAM error will be small too. This underlying assumption was taken for granted by previous works without any justification. However, in the light of the above error decomposition, this assumption is not right. The reconstruction error is only part of the total error. To minimize the total error, we should not only consider the reconstruction error, but also the search error.

As the residual energy of a PCA subspace is linearly related to the percentage of energy retained, the reconstruction error is linearly related to the explanation proportion , i.e. $ERR_{REC} = (1-\alpha)\sum\lambda_i$. However, the search error is rather more complex. It is determined both by the search algorithm and the noise distribution on the data: first, the AAM search algorithm with the prediction matrices is not accurate; second, noise on the data can also disturb the search procedure. Due to its complexity, a practical way is to learn it from the data as done in the following sections.

5. Optimizing Subspaces in AAM

Now we will use the error decomposition to find the optimal subspaces in AAM. We first develop a new criterion for selecting eigenvectors for any subspace, and then present a method to implement it. Finally, another method is proposed to find all the optimal subspaces in AAM.

5.1. Criterion for Selecting Eigenvectors

To obtain the optimal P-$\mathbb{S}$, we try to find out what criterion should be met for the eigenvectors in P-$\mathbb{S}$. In order that the P-$\mathbb{S}$ containing the first t eigenvectors are optimal, we should have:

$$\begin{cases} ERR^{t}_{(x)} < ERR^{t-1}_{(x)} \\ ERR^{t}_{(x)} < ERR^{t+1}_{(x)} \end{cases} \tag{10}$$

That is:

$$\begin{cases} (x'_t-y'_t)^2 < (x_t)^2 \\ (x'_{t+1}-y'_{t+1})^2 > (x_{t+1})^2 \end{cases} \tag{11}$$

For all the target objects, we will have:

$$\begin{cases} E[(x'_t-y'_t)^2] < E[(x_t)^2] \\ E[(x'_{t+1}-y'_{t+1})^2] > E[(x_{t+1})^2] \end{cases} \tag{12}$$

Note that we do not assume any order about the eigenvectors in this derivation. The first t eigenvectors do not have to be the ones with the first t biggest eigenvalues. So it means that for the optimal P-$\mathbb{S}$, any eigenvector i in P-$\mathbb{S}$ should satisfy:

$$E[(x'_i-y'_i)^2] < E[(x_i)^2] \tag{13}$$

And any eigenvector j in the C-$\mathbb{S}$ should satisfy:

$$E[(x'_j-y'_j)^2] > E[(x_j)^2] \tag{14}$$

In other words, any eigenvector i belongs to the optimal P-$\mathbb{S}$ if and only if it satisfies:

$$E[(x'_i-y'_i)^2] < E[(x_i)^2] \tag{15}$$

5.2. Practical Method for Selecting Eigenvectors for One Subspace

According to the selection criterion of the eigenvectors for the optimal P-$\mathbb{S}$, Equation (15), the ideal way is to test all the possible combinations of the eigenvectors and select the optimal combination as the optimal P-$\mathbb{S}$. But the number of all the possible combinations is too big to do all the tests. For example, for the shape space with 87 points, the number of the possible combinations of the eigenvectors is $\sum_{i=1}^{170}\binom{170}{i}=2^{170}-1$.

In this paper, we propose a practical method, which is both efficient and effective, to select the eigenvectors. This method is based on the observation that the search error distribution for each eigenvector is fairly steady and does not vary too much with different combinations of the eigenvectors. Therefore, we can use the error distribution of the model containing all the eigenvectors to approximate any combination of them. This method works as follows:

1. First, use the P-$\mathbb{S}$ with a high proportion (e.g. 0.99) of all the eigenvectors for searching.

2. Then, select eigenvectors belonging to the optimal P-$\mathbb{S}$ according to the criterion Equation (15).

In practice, after getting the optimal P-$\mathbb{S}$, we can still use a slightly higher explanation proportion than the optimal one to test for the final optimal P-$\mathbb{S}$.

5.3. Optimizing All the Subspaces in AAM

There are three subspaces in AAM, i.e. the shape subspace $\mathbb{S}_s$, the texture subspace $\mathbb{S}_t$ and the appearance subspace $\mathbb{S}_a$. To find the optimal subspaces is to find three explanation proportions: α_s for $\mathbb{S}_s$, α_t for $\mathbb{S}_t$ and α_a for $\mathbb{S}_a$. The optimal subspaces are a combination of (α_s, α_t, α_a). This is a three parameter optimization problem, traditional cross-validation is hard to be implemented. But with the method in the above subsection, the optimal subspaces can be found in the following way:

1. First, use the above method to find the optimal subspaces for the shape and texture.
2. Then, use the above method to find the optimal appearance subspace constructed from the optimal subspaces of shape and texture.

6. EXPERIMENTS

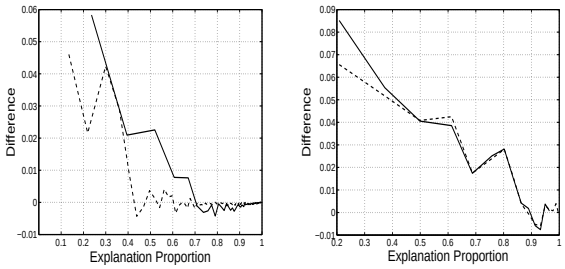
The database used consists of 200 face images from the FERET [13], the AR [12] databases and other collections. 87 landmarks are labelled on each face. We randomly select 100 images as the training and the other 100 images as the testing images. Our AAM implementation is based on AAM-API [2] [14].

6.1. Optimizing Subspaces for AAM

On each test image, we initialize AAM with displacements from the true position by ± 10 pixels in both x and y (4 initializations). Note that different initializations may cause different optimal explanation proportions as their noise distributions are different. We choose this initialization so that most of the searches will converge to the target. Our task is to find the optimal combination of $(\alpha_s, \alpha_t, \alpha_a)$

We first train the AAM model with the combination of $(0.99, 0.99, 0.99)$ to find optimal values for α_s and α_t. We plot $E[(x_i)^2] - E[(x'_i - y'_i)^2]$ for all the explanation proportions of the shape and texture eigenvectors in Figure 2(a), in which solid line denotes the shape subspace and dotted line denotes the texture subspace. According the selection criteria, Equation (15), we should choose the eigenvectors with $E[(x_i)^2] - E[(x'_i - y'_i)^2] > 0$, corresponding to the curve points above zero. So the optimal values for α_s and α_t are $(0.66, 0.59)$ corresponding to the first 5 and 13 eigenvectors in the shape and texture space respectively. Then we train the AAM model with $(0.66, 0.59, 0.99)$ to find the optimal value for α_a. $E[(x_i)^2] - E[(x'_i - y'_i)^2]$ for all the explanation proportions is shown in Figure 2(b) with dotted line and the optimal value for α_a is 0.86, corresponding to the first 9 eigenvectors in the appearance space. Note that there are some burrs or noises on the curves. But they are not much enough to disturb the results.

To test the robustness of estimating α_s, α_t, we also trained the AAM model with $(0.98, 0.98, 0.98)$ and $(0.95, 0.95, 0.95)$. They give the same results as except that there are more burrs for the latter. To test the robustness of estimating α_a, we train the AAM model with $(0.66, 0.59, 0.98)$ and $(0.66, 0.59, 0.95)$. The former gives the same results as $(0.66, 0.59, 0.99)$, but the latter gives $(0.66, 0.59, 0.88)$, which is shown in Figure 2(b) with solid line.



(a) Shape and Texture (b) Appearance

Figure 2. $E[(x_i)^2] - E[(x'_i - y'_i)^2]$ **for different explanation proportion of the shape, texture and appearance subspaces**

6.2. Performance Comparison for different Subspaces

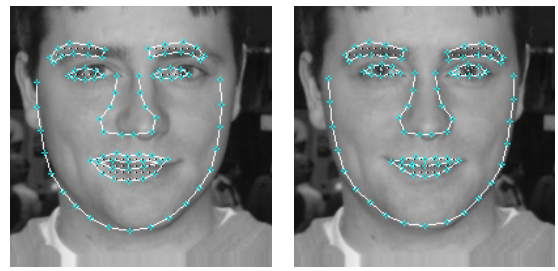
Initializations are the same as the previous subsection. The searching results are compared with the labelled shapes. We use point-to-point error(Pt-Pt), point-to-boundary error(Pt-Crv) (the distance from the found points to the labelled boundaries) as the comparison measures. The comparison results are shown in Table 1. The results of parallel analysis (PA)[14] are also given. PA explanation proportions correspond to the first 7, 21 and 4 eigenvectors in the shape, texture and appearance space respectively. We can see that even if the optimal explanation proportion is not the best, it is very near to the best results. One typical searching example is given in Figure 3. We can see that the model with the explanations of 0.98 is too flexible so that it is easy to be stuck in lo-

2 Available: http://www.imm.dtu.dk/∼aam/

cal minimum(mouth and nose). However, the model with PA explanations is too strict so that it can not fit accurately to the target.

α_s	α_t	α_a	Pt-Pt (Pixels)	Pt-Crv (Pixels)	Time (ms)
0.66	0.59	0.86	2.71	1.62	631
0.66	0.59	0.88	2.69	1.60	726
0.66	0.62	0.86	2.69	1.59	737
0.70	0.59	0.86	2.69	1.59	710
0.95	0.95	0.95	3.23	1.75	2327
0.98	0.98	0.98	3.38	1.81	3371
0.73 (PA)	0.70 (PA)	0.55 (PA)	3.15	1.88	414

Table 1. Comparative performance of different subspaces



(a) Initialization (b) Optimal subspaces

(c) PA subspaces (d) 0.98 subspaces

Figure 3. Example for AAM with different subspaces

7. Conclusion and Future Work

In this paper, we presented an error analysis of the Active Appearance Models (AAM) by decomposing the AAM subspace errors into the reconstruction errors and the search errors, and then devised a method to find the optimal AAM subspace model with respect to the search procedure. This method considered both the subspace model and the search procedure. This method can be used to optimize other alignment algorithms, such as ASM and AAA [1]. Future work includes applying this method to variations of AAM and other alignment algorithms.

References

[1] J. Ahlberg. Using the active appearance algorithm for face and facial feature tracking. In *IEEE ICCV Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, pages 68–72, Vancouver, Canada, July 13 2001.

[2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090–1097, Hawaii, December 2001.

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV98*, volume 2, pages 484–498, 1998.

[4] T. F. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. In *Proceedings of the British Machine Vision Conference*, volume 2.

[5] T. F. Cootes and C. J. Taylor. Constrained active appearance models. In *Proceedings of IEEE International Conference on Computer Vision*, pages 748–754, Vancouver, Canada, July 2001.

[6] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. Technical report, `www.isbe.man.ac.uk/~bim/refs.html`, 2001.

[7] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *CVGIP: Image Understanding*, 61:38–59, 1995.

[8] G. J. Edwards, T. F. Cootes, and C. J. Taylor. Interpreting face images using active appearance models. In *Proc. International Conference on Automatic and Gesture Recognition*, pages 300–305, Japan, 1998.

[9] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285–339, 1991.

[10] X. Hou, S. Li, and H. Zhang. Direct appearance models. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, number 1, pages 828–833, Hawaii, December 2001.

[11] M. J. Kearns, Y. Mansour, A. Y. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27:7–50, 1997.

[12] A. Martinez and R.Benavente. The AR face database. Technical Report 24, CVC, June 1998.

[13] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[14] M. B. Stegmann, B. K. Ersboll, and R. Larsen. FAME - a flexible appearance modeling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319–1331, October 2003.