# Person-Independent Face Tracking Based on Dynamic AAM Selection

Akihiro Kobayashi
National Institute of Information and Communication Technology

Junji Satake
National Institute of Information and Communication Technology

3–5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan

Takatsugu Hirayama
Kyoto University

Hiroaki Kawashima
Kyoto University

Takashi Matsuyama
Kyoto University

Yoshida-Honmachi, Sakyo-ku Kyoto 606-8501, Japan

## Abstract

*We have developed a high-precision method that selects an appropriate model of a video image in order to track an unknown face in front of a large display. Currently, Active Appearance Models (AAMs) are used to track non-rigid objects, such as a faces, because the models efficiently learn the correlation between shape and texture. The problem with an AAM is that when it tracks an unknown face, excessive training data increases tracking errors because there is an intermediate model size beyond which the reduction in fitting performance outweighs the gains from any improved representational power of the model. To increases the accuracy with which an unknown face is tracked, we built clustered models from training datasets and select a cluster that includes a face which is similar to the unknown face. Our method of clustering and cluster selecting is based on the Mutual Subspace Method (MSM). We demonstrated the effectiveness of our method by using the leave-one-out cross-validation.*

## 1. Introduction

We developed an interactive information display system that also reads a user's unconstrained non-verbal behavior, such as face and gaze direction, in order to guess such things as the user's real intentions or true preferences. Figure 1 shows the system. The system has a large display and multiple cameras, and interactively shows information reacting to the user's intention and preference. This system can be applied to situations that involve a dialogue, such as making travel plans or explaining items of interest to museum visitors.

Reading a user's non-verbal behavior depends on a system being able to accurately estimate face and gaze direction; therefore, facial tracking technology, which performs this function, is essential. An Active Appearance Model (AAM) [3] is often used for facial parts tracking, such as the analysis of facial expressions [7]. Using an AAM can provide fast and stable non-rigid object tracking because it is a statistical model that shows the correlation between shapes (coordinate values of feature points) and grey-levels (intensity of each pixel). However, when an AAM is used for person-independent face tracking, excessive training data decreases tracking accuracy as we will describe in the next section. To improve the accuracy of tracking an unknown user, we use a clustered model made from face images that are similar to the user's face image. Similar individuals from face database containing a large number of individual face data are found and merged using a hierarchical cluster analysis. Finally, we dynamically select the most appropriate cluster for the user and used it to provide highly accurate face tracking.

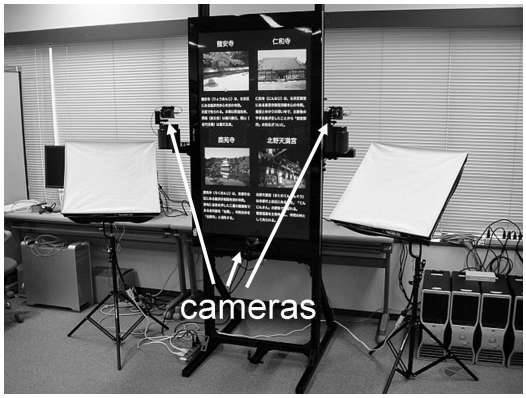



Figure 1. Information display system

978-1-4244-2154-1/08/$25.00 

## 2. Related Works

Using an AAM involves performing a PCA-subspace of face images, as does using many other face tracking methods. For example, Sparse Eigentemplate [10] uses face subspace with Condensation and tracks human faces stably. However, Sparse Eigentemplate can track only trained faces.

Some traditional approaches to extracting an unknown face from images involve training a model by using many individual images in order to improve the ability of the subspace to express the variety of human faces [13, 14]. In methods involving facial parts tracking, there is some research on making an individual eigenspace for each facial part, such as the eye, nose, and mouth, from many individual images in order to detect facial parts in images [8, 17]. However, using excessive training data to train a subspace decreases the tracking accuracy because there is an intermediate model size beyond which the reduction in fitting performance outweighs the gains from the model's improved representational power [4]. Using a simultaneous inverse compositional algorithm increases the fitting performance. The algorithm fits shapes into a face image by performing a Gauss-Newton gradient descent optimization simultaneously on the warp parameters and the appearance parameters. The algorithm, however, needs high computational power and cannot work in real-time [4].

To solve the problem of reduced tracking accuracy, we dynamically change a clustered model into the most appropriate one made from similar face images. In research that does not involve eigenspace, a tracking method developed by Sugano et. al. [11] and a tracking method called Elastic Bunch Graph Matching (EBGM) [16] change models to deal with the differences between individual faces and can be used to accurately detect facial parts. Sugano et. al. uses incremental bundle adjustment for person-dependent shape estimation. In EBGM, a system takes many Gabor features, and selects an appropriate one for an input image. We try to build clusters that have an appropriate size so that they can express an unknown individual face as a subspace of the cluster.

## 3. Tracking Based on AAM [3]

### 3.1. Learning Procedure

Cootes et. al. proposed an efficient method to learn the correlation between shape and texture [3]. Initially, gray-level variance independent from shape variance is needed for learning the correlation between shape and grey-level. The training data for an AAM is a set of images and coordinate values of feature points on the images. Figure 2 shows an example of training data recorded by the system in Figure 1. We put 45 feature points on the image. In this paper,

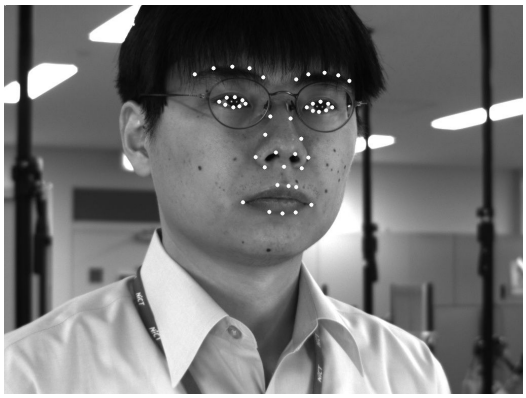


Figure 2. Training data for AAM

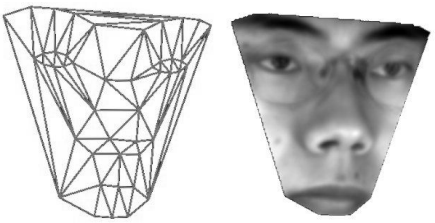


Figure 3. A mesh for warp and a warped image

a vector composed of coordinate values on feature points is called *shape vector* $s$. To build an AAM, rotation and translation of shape vectors of training sets are normalized. The face region is extracted from an image along its feature points, and its shape is normalized into a mean shape $\overline{s}$ of the normalized shapes. This process is called *Warp*. Piecewise affine or thin plate spline can be used for this normalization [2, 9]. We use piecewise affine based on a mesh composed of the feature points. Figure 3 shows a mesh and an image warped by the mesh. A vector composed of intensity values in the warped image is called a *grey-level vector* $g$.

Next, the distribution and correlation between shapes and grey-level is calculated . When PCA is performed on a set of shape vectors $s$ and grey-level vectors $g$ in training data, Equation (1) gives approximations of $s$ and $g$:

$$s = \overline{s} + U_s c_s \ , \ g = \overline{g} + U_g c_g \tag{1}$$

where $\overline{s}$ is a mean vector of $s$, $\overline{g}$ is a mean vector of $g$, $U_s$ and $U_g$ are orthogonal matrixes where each column vector is a base vector, and $c_s$ and $c_g$ are coefficients of basis vector. Since there may be correlations between the shape and grey-level variations, both of the vectors are concatenated into a vector, and PCA is performed on the vector as follows.

$$\begin{bmatrix} W_s c_s \\ c_g \end{bmatrix} = c = \begin{bmatrix} V_s \\ V_g \end{bmatrix} d = V d \tag{2}$$

(a) Model from 1 User (b) Model from 6 Users

Figure 4. Tracking result based on personal / multi-users' AAM

where $W_s$ is a diagonal matrix of weights for each shape parameter, allowing for differences in units between the shape and grey-level, $V$ is a set of orthogonal models, and $d$ is a parameter vector controlling both the shape and grey-levels of the model. Note that the linear nature of the model allows us to express the shape vector $s$ and grey-level vector $g$ directly as a function of $d$

$$s = \overline{s} + U_s {W_s}^{-1} V_s d \ , \ g = \overline{g} + U_g V_g d \tag{3}$$

where $V = ({V_s}^T, {V_g}^T)^T$. In other words, an example image can be synthesized for a given $d$ by generating the shape-free grey-level image from the vector $g$ and warping it using the feature points described by $s$.

### 3.2. Searching Procedure

If a new image and the model $(U_s, W_s, V_s, U_g, V_g)$ are given, we can treat face tracking as an optimization problem in which we minimize the grey-level difference between a new image and a synthesized image using the parameter vector $d^*$.

$$d^* = \arg\min_d |g_u - g_v|^2 \tag{4}$$

where $g_v$ is a synthesized image projected from parameter $d^*$ by Eq. (3), and $g_u$ is a new image warped by a candidate of optimized shape $s*$ projected from $d^*$ by Eq. (3). $|g_u - g_v|$ are iteratively minimized. In most previous algorithms, it was simply assumed that there was a constant linear relationship between this error image [1] and the additive incremental updates to the parameters $d^*$ [3].

### 3.3. Problems with AAM

When we use an AAM for person-independent face tracking, there is an inherent trade-off between performance to express previously unseen data and performance to fit those data. A generic model based on large number of individual datasets has not only power of representation, but also difficulty of fitting. The main reason for the difficulty appears to be that the effective dimensionality of the generic shape model is far higher than that of the person-specific shape models [4].

Figure 4(a) shows the result of tracking a user by using an AAM trained from the user. Figure 4(b) shows the result of tracking the user by using an AAM trained from the user and additional 5 users. Figure 4(b) shows less accuracy than 4(a). There is a significant mismatch in the left eye in 4(b) because the result of 4(b) falls into a local minima. To solve this problem, we dynamically change clustered models into the most appropriate one made from faces those are similar to a given user's face.

## 4. Build and Select Cluster

### 4.1. Distance Between Clusters

To select the most similar cluster from input images or cluster training sets, we define similarities between clusters. We utilize inter-cluster similarities defined by using the Mutual Subspace Method (MSM) [6], which is applied to facial recognition [17]. Thanks to the use of the MSM, a variety of facial expressions and directions can be dealt with because the distributions of video images are exploited in the MSM; meanwhile, the still images are used in traditional methods. Figure 5 shows the concept of the MSM. In the MSM, similarity is measured between two subspaces, $\mathcal{L}_1$ and $\mathcal{L}_2$, based on the smallest canonical angle, $\theta_1$, between $\mathcal{L}_1$ and $\mathcal{L}_2$. Using the MSM, the similarity of these subspaces is defined as

$$\cos^2\theta_1 = \max_{u \in \mathcal{L}_1, v \in \mathcal{L}_2, ||u|| \neq 0, ||v|| \neq 0} \frac{|(u,v)|^2}{||u||^2 ||v||^2} \tag{5}$$

where $u$ and $v$ are vectors on the subspace while Equation (5) has local maxima.

Maeda et. al. proposed a fast method to calculate the smallest canonical angle $\theta_1$ based on projective matrixes [6]. $\mathbf{P_1}$ and $\mathbf{P_2}$ are $F \times F$-dimensional projective matrixes from $F$-dimensional image space $\mathcal{V}$ projected onto subspace $\mathcal{L}_1$ and $\mathcal{L}_2$. $\mathbf{P_1}$ and $\mathbf{P_2}$ are defined as

$$\mathbf{P_1} = \sum_{i=1}^{M} \Phi_i {\Phi_i}^T, \mathbf{P_2} = \sum_{i=1}^{N} \Psi_i {\Psi_i}^T \tag{6}$$

where $\Phi_i$ and $\Psi_i$ are base vectors of subspace $\mathcal{L}_1$ and $\mathcal{L}_2$. By defining the canonical angle, $\cos^2\theta_1$ is equal to the largest eigenvalue of $\mathbf{P_1 P_2}$ or $\mathbf{P_2 P_1}$. Maeda decreased the number of dimensions for the calculations, based on the theorem; $\cos^2\theta_1$ is also equal to the largest eigenvalue of $\mathbf{P_1 P_2 P_1}$ or $\mathbf{P_2 P_1 P_2}$.

[1] the right side of Equation (4)

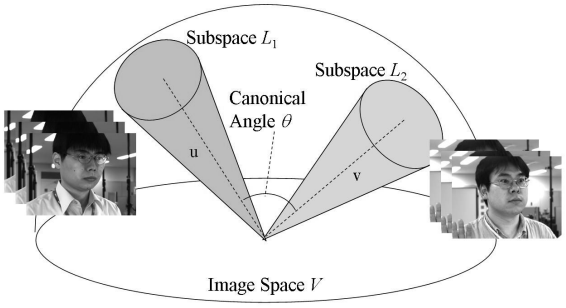



Figure 5. Mutual Subspace Method

### 4.2. Build Hierarchy of Clusters

We analyze training datasets in the facial database and built hierarchical clusters of training data in order to get an appropriate set of AAMs. In the training face, our system builds a subspace for each individual. We call the subspace *individual subspace*. The system applies a hierarchical clustering algorithm [5] to the individual subspaces, as follow.

**step 0** Define the distance between individual subspaces as $1-cos^2\theta_1$. ($\theta_1$ is the smallest canonical angle between these subspaces.)

**step 1** Start by assigning each individual subspace to a cluster.

**step 2** Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.

**step 3** Compute distances (similarities) between the new cluster and each of the old clusters.

1. The system rebuilds a new subspace of the merged cluster and computes a new canonical angle between the new subspace and the others.
2. The system defines distance between clusters based on individual subspaces.

**step 4** Repeat steps 2 and 3 until all items are clustered into a single cluster.

At step 3, there are two types of methods for computing the distance between clusters. If the system rebuilds a new subspace, a new merged cluster is often merged with the other at the next step because distances between the new cluster and the others often become shorter than that between old clusters. In consequence, a chain like dendrogram is built. To avoid this problem, we use individual subspaces, and Ward's method [15], which is the hierarchical cluster analysis, because the Ward's method uses an analysis of variance approach to evaluate the distances between clusters. In short, this method attempts to minimize the sum of squares (SS) of any two clusters that can be formed at each step. In general, this method is regarded as very efficient; however, it tends to create small clusters.

### 4.3. Overview of Selecting Process

We build a set of subspaces from grey-level vectors $g$ warped by feature points from face images and used distances between the subspaces for clustering training datasets.When we compare these subspaces, we can easily find the difference between individuals because the vectors $g$ are given from warped image by a mean shape and the vectors suppresses inner variations of shapes in an individual, such as facial expression, as described in Section 3.1. Figure 6 shows an overview of building and selecting a cluster. As shown in this figure, our system has a facial database, which has training datasets to build AAMs. The facial database has a hierarchical structure classified by the algorithm described in Section 4.2. For instance, let the labels *X-Z* denote the clusters in a layer of the hierarchy. To track a face and select a model, we compute (a)-(d).

(a) AAM trained by all available data (for pre-tracking)
(b) AAM trained by each cluster (for tracking)
(c) Subspace of each cluster (for selecting cluster)
(d) Subspace of input data (for selecting cluster)

We assume that the system trains (a)-(c) from the facial database and builds online (d) from input images. Initially, when selecting a model, the system roughly pre-tracks the input image using (a). Next, the system extracts the face along the feature points, and warps the face image along a mean shape of pre-tracked results. The system applies K-L expansion to the warped image and gets a subspace (d). Finally, the system determines the similarity between (c) and (d) for each cluster, selects the most similar cluster, and then accurately tracks the input image using (b) which is included in the cluster.

We would develop a simple system if we could compare AAMs directly; however, it is difficult to determine similarity between AAMs. That is, each image space of each AAM (b) is independent because each image space is warped by each mean shape $\overline{s}$, as shown in Figure 3. As shown in Figure 5, it is not possible using the MSM to compare AAMs directly because it is assumed that subspace $\mathcal{L}_1$ and $\mathcal{L}_2$ are included in the same image space. We build (b) and (c) independently because the system can track a user more accurately based on AAM (b) built with the mean shape of the most appropriate cluster for the user, rather than based on (b) built with the mean shape of all the clusters. In Section 6.3, we discuss how to integrate (b) and (c).

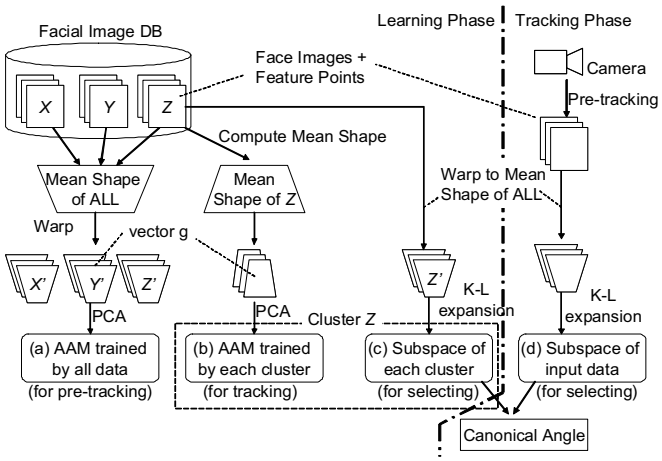



Figure 6. Overview of subspace selection

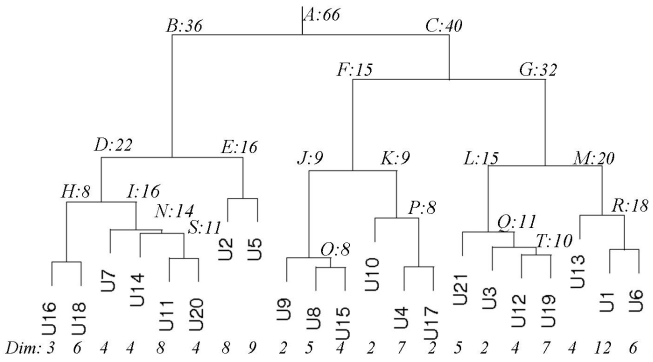



Figure 7. Result of clustering 21 individual-subspaces

## 5. Experiment

### 5.1. Evaluation Dataset

At first, we built an experimental environment to get evaluation datasets. We used one of the three cameras attached around the plasma display panel (PDP) in Figure 1; the camera is located on the right side of the display from users' viewp(oint. The camera is Dragonfly2 (XGA, 30fps, 8bit gray image, 1/3 inch CCD), made by Point Grey Research Inc. We attached a lens, HF12.5HA-1B (f=12.5mm) made by FUJINON Inc. to the camera. To keep the lighting environment stable, two lights are set, as shown in Figure 1.

Next, we recoded and built evaluation datasets. We recorded video images of 21 examinees (14 males, 7 females) aged between 20 and 50. Each examinee stood 1 meter away from the PDP. The PDP displayed 15 markers (width 3 $\times$ height 5) at 20cm intervals. Each examinee sequentially looked at each of the markers directing their face to the markers consciously. Examinees who wore glasses took their glasses off. We built clusters, which is a set of a subspace and an AAM as described in Section 4.3, from the videos. We picked up one frame that showed the examinee looking at each marker, giving a total of 15 frames for each examinee. We manually put 45 feature points on each frame. Figure 2 shows an example of images which our system got, and shows feature points which we manually put. We use them as training data and ground trouth of test sets in this expreriment.

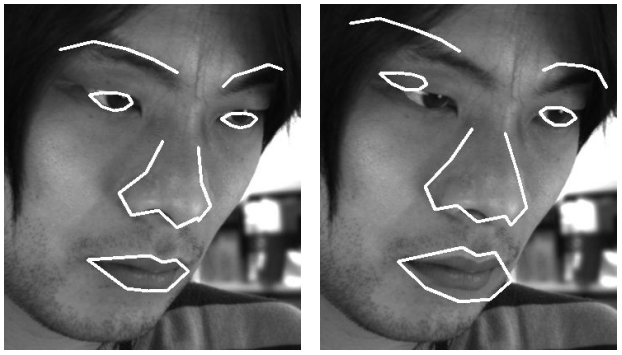
Finally, we built these clusters. We built each individual subspace based on the 15 frames, as described at Section 4.3. The dimension of a subspace was defined, so that proportion of variance became over 95%. We then obtained the distance between individual subspaces by using the MSM and applied the hierarchical clustering analysis shown in Section 4.2. To build these clusters, we used R [1], a free software environment for statistical computing. Figure 7 shows a dendrogram analyzed by R. Labels start from "U" denote user IDs, labels *A-T* denote cluster IDs, and the numbers in italics are the dimensions of each cluster.

### 5.2. Evaluation of Tracking Accuracy

We estimated tracking accuracy using the leave-one-out cross-validation. We removed data of one examinee as a test set from each of obtained clusters in Figure 7 and re-built the AAM except for the test-set. We tracked test-set videos using AAMs of the 1st layer (*B, C*) and 2nd layer (*D, E, F, G*). At first, we refer a cluster that includes a test-set as a *cluster* $\alpha$. For example, when user 12 is test-set, we call *C* and *G* a *cluster* $\alpha$. In this experiment, the test-set is removed from the cluster $\alpha$ befor building an AAM and a subspace for leave-one-out cross-validation. For example, when user 12 is test-set, we built two clusters as cluster $\alpha$. The cluster $\alpha$ on the 1st layer is built from user 9, 8, 15, 10, 4, 17, 21, 3, 19, 13, 1, and 6. The cluster $\alpha$ on the 2nd layer is built from user 21, 3, 19, 13, 1, and 6.

Next, we refer a cluster, which is on the same layer of the cluster $\alpha$ but not includes the test-set, as a *cluster* $\beta$ to compare tracking error. For example, when test-set is user 12, we compared *C* to *B* as $\alpha$ and $\beta$ in the 1st layer, and compared *G* to *F* as $\alpha$ and $\beta$ in the 2nd layer. In other words, a cluster $\alpha$ includes similar faces with a test-set, and the cluster $\alpha$ should be selected to track the test-set in the process shown in Figure 6. For example, Figure 8 was clipped from a XGA image which shows the result of tracking. Figure 8(a) shows the result of tracking user 12 using an AAM of cluster *C* (cluster $\alpha$), and Figure 8(b) shows that of cluster *B* (cluster $\beta$).

We defined feature points put manually into training data as ground truth. We compared the ground truth with the tracking results of clusters $\alpha$ and $\beta$. Table 1 shows mean errors of 45 feature points in 15 frames, which were used as



(a) $\alpha$: Tracking U12 Using Cluster *C* without U12 data

(b) $\beta$: Tracking U12 Using Cluster *B*

Figure 8. Tracking result based on AAM from cluster $\alpha$ / $\beta$

a training set. The mean error M was defined as follows.

$$M = \frac{1}{nl} \sum_{j=1}^{l} \sum_{i=1}^{n} sqrt[(x_i - x'_i)^2 + (y_i - y'_i)^2] \quad (7)$$

where $n$ is the num of feature points, and $l$ is the num of frames. $(x'_i, y'_i)$ shows position of feature point $i$ on image coordinate system as a ground trouth, and $(x_i, y_i)$ shows that of tracked point.

In Table 1, we colored failed cases; i.e. we colored cells of the test sets that the error of cluster $\alpha$ was larger than that of $\beta$. We counted the number of cases when the error of $\alpha$ was smaller than that of $\beta$. In the 1st layer, 16 tests out of 21 tests (76.2%) corresponded to this case. In the 2nd layer, 17 tests out of 21 tests (81.0%) corresponded to this case. This result shows that selecting a cluster built from similar individuals decreases tracking errors. It also shows the dendrogram built in Section 4.2 is valid.

### 5.3. Evaluation of Selecting a Cluster

We tested whether our system can select the most appropriate cluster using the performance evaluation in the above section. We predict that performance using the leave-one-out validation was the same as in Section 5.3. We built a subspace of a cluster in Figure 6 after removing a test-set from the cluster. We defined the test-set as input images and built a subspace from the input image. Finally, we computed the similarity between input images and each cluster. We did 21 trials for each user to select a cluster from the 1st layer (*B*-*C*) and 2nd layer (*D*-*G*). In each trial, we compared the similarity of cluster $\alpha$ to the similarity of cluster $\beta$. We defined the trial as a success when input images has larger similarity to cluster $\alpha$ than cluster $\beta$.

Table 2 shows the number of successful trials. We defined the dimension of subspaces in two ways in this experiment. In the first way, we defined the dimension so that proportion of variance became over 95%. In the second way, we used constant dimensions (top 5 axes). As shown in Table 2, in the second experiment using constant dimensions, 81.0% of trials succeeded in the 1st layer and 85.7% of trials succeeded in the 2nd layer. The 2 cases of 3 failed trials were for user 2 and user 5. In these cases, the similarity of cluster *E* tended to be small because cluster *E* had only single individual data after the leave-one-out validation, as shown in Figure 7. The results demonstrated that our system can select a cluster that brings more accurate tracking if the cluster has enough datasets.

In the first experiment using constant proportion of variance, the system tended to select a cluster that had more datasets. In the MSM, a canonical angle between large clusters tend to become small, because a larger cluster has better peformance of representation. On the other hand, the definition based on proportion of variance is good for tracking because the definition shows enough axes to express its image space. If the facial database has enough data, clusters

Table 1. Mean tracking error [pixel] using cluster $\alpha$ or $\beta$. *Cluster* $\alpha$ is a cluster that includes a test-set. *Cluster* $\beta$ is a cluster on the same layer of the cluster $\alpha$ but not includes the test-set. A colored cell shows a failed case where tracking error of cluster $\alpha$ is larger than that of $\beta$.

| test set | 1st layer | | 2nd layer | |
|---|---|---|---|---|
| | $\alpha$ | $\beta$ | $\alpha$ | $\beta$ |
| 1 | 8.14 | 12.08 | 6.78 | 12.86 |
| 2 | 8.87 | 10.99 | 10.98 | 9.03 |
| 3 | 5.45 | 5.38 | 5.77 | 5.92 |
| 4 | 6.38 | 6.30 | 6.93 | 7.10 |
| 5 | 9.20 | 18.88 | 10.16 | 16.27 |
| 6 | 9.07 | 9.94 | 7.21 | 12.29 |
| 7 | 5.82 | 7.98 | 6.11 | 9.03 |
| 8 | 6.16 | 7.60 | 6.18 | 9.89 |
| 9 | 6.41 | 7.89 | 6.38 | 7.37 |
| 10 | 7.46 | 9.61 | 8.11 | 8.00 |
| 11 | 8.27 | 11.61 | 9.30 | 10.39 |
| 12 | 6.27 | 9.01 | 8.28 | 8.51 |
| 13 | 10.00 | 9.80 | 10.85 | 12.70 |
| 14 | 5.54 | 13.71 | 7.04 | 9.61 |
| 15 | 8.00 | 7.56 | 11.18 | 10.57 |
| 16 | 4.87 | 5.47 | 4.98 | 7.31 |
| 17 | 4.26 | 5.72 | 4.67 | 5.44 |
| 18 | 7.23 | 7.89 | 10.10 | 8.54 |
| 19 | 6.75 | 6.52 | 6.77 | 8.24 |
| 20 | 6.63 | 10.67 | 5.39 | 8.10 |
| 21 | 4.51 | 5.71 | 4.97 | 5.35 |

Table 2. Rate of trials selecting appropriate cluster. We counted the trials when input images has larger similarity to cluster $\alpha$ which includes the test-set images than cluster $\beta$ in the 1st layer or 2nd layer of the dendrogram shown in Figure 7.

| layer | 1st layer | 2nd layer |
|---|---|---|
| constant proportion | 81.0% | 71.4% |
| constant dimension | 100.0% | 85.7% |

Table 3. Mean tracking error using an each level cluster of the dendrogram shown in Figure 7

| cluster | A | C | G | L | Q | T |
|---|---|---|---|---|---|---|
| error | 6.41 | 6.27 | 8.28 | 8.43 | 11.75 | 16.44 |

in the same layer will have same number of dimensions because the dimensions of the subspace will be saturated. In the selecting process, however, we need to consider the difference of dimensions between clusters.

In this paper we built clusters before removing test set, in order to check the both performance, accuracy (Sec. 5.3) and selection (Sec. 5.3). Strictly speaking, we should build clusters after removing test set, in order to evaluate our method in practical environment. If we build clusters after the leave-one-out, our system will show the same tendency toward this experiment because experiments of Section and demonstrated that a model which gave more accurate tracking tended to be selected.

# 6. Discussion

## 6.1. Best Size of Clusters

To investigate the most appropriate layer to track a user, we compared the tracking error for user 12 based on a cluster of each layer (*A, C, G, L, Q, T*) using the leave-one-out validation, as described in Section 5.3. Table 3 shows the mean error. In the layers from clusters *C* to *T*, the larger cluster brings more accurate tracking. However, the tracking error using cluster *A* is larger than cluster *C*, as shown in Table 3. Cluster *A* uses AAM trained on the entire dataset without user 12. This means that the comparison between *C* and *A* shows comparison between the tracking error the model selection technique and that of a standard AAM trained on the entire dataset. In this case, the result of the comparison shows the effectiveness of our selection technique.

Figure 7 shows the reason of this result. The number of dimensions of the clusters linearly increases from *T* to *C* [2]. However, the rate of increasing dimensions becomes down between *C* and *A*. This means that the reduction in fitting performance outweighs the gains from any improved representational power of the model. In many other tests with different users, however, the most accurate cluster became *A*. The most likely reason for this is that there were not enough datasets to saturate dimensions. Therefore, if there is enough data in the facial database, a clustered AAM obtained from partial datasets would become the most accurate model, as seen in Table 3.

[2]Dimension of a cluster approximates total number of dimensions of its children.

## 6.2. Effect of Pre-Tracking Errors

Our system applies the MSM to results of the pre-tracking based on an AAM trained by all available data ((a) in Figure 6). Any error in the pre-tracking would cause an error in selecting clusters. However, our selecting method can be less prone to tracking errors because the AAM uses texture-based matching, as described by equation (4). In addition, when we will implement our method on an online-system, we can deal with this problem by changing cluster layers depending on the tracking accuracy.

## 6.3. Integration of AAM and MSM

As described in Section 4.3, we built an AAM to independently from a subspace in order to implement accurate tracking. If an AAM for each cluster is warped into the mean shape of all available data, we can directly compare the AAMs and perform cluster analysis on them. The process to warp a individual face into the mean shape decreases the accuracy of tracking. In this case, an AAM needs to be used in combination with some method of partial matching, such as the active shape model developed by Sung [12], in order to deal with tracking errors.

## 6.4. Clustering Based on Face Directions

This paper attends to differences of appearance between individual faces. We defined an individual subspace as a minimum unit.However, appearance of an individual face has large differences when it directs different directions. When the angle between a camera and face is large, warp cannot normalize the difference using these facial directions, and therefore there are often tracking errors. If our system builds a cluster based on facial directions, we will be able to track a face more accurately.

# 7. Conclusion

We increased the accuracy of AAM-based tracking an unknown person by selecting clusters of similar individuals based on the MSM. We estimated tracking errors for each cluster by the leave-one-out validation. Using our method, it is possible to select appropriate clusters for each input image. Finally, we demonstrated that using our method was effective by showing a clustered AAM was more accurate than an AAM based on all available data, and discussed the appropriate size of clusters. In the future, we will improve the way clusters are selected, will implement an online system, and will demonstrate the efficiency of our method using a large number of facial data base.

# Acknowledgment

This work is in part supported by Grant-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports,

Science and Technology of Japan under the contract of 18049046.

## References


[1] http://www.r-project.org/.

[2] S. Baker, R. Gross, and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 3, technical report cmu-ri-tr-03-35. Technical report, Carnegie Mellon University Robotics Institute, 2003.

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

[4] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. In *Image and Vision Computing*, volume 23, pages 1080–1093, 2005.

[5] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, (2):241–254, 1967.

[6] K. Maeda and S. Watanabe. A pattern matching method with local structure. *IEICE Trans. Inf. and Syst. (Japanese Edition)*, J68-D(3):345–352, 1985.

[7] M. Nishiyama, H. Kawashima, T. Hirayama, and T. Matsuyama. Facial expression representation based on timing structures in faces. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (W. Zhao et al. (Eds.): AMFG 2005, LNCS 3723)*, pages 140–154, 2005.

[8] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. CVPR*, 1994.

[9] S. Sclaroff and J. Isidoro. Active blobs. In *Proc. ICCV*, pages 1146–1153, 1998.

[10] T. Shakunaga, Y. Matsubara, and K. Noguchi. Appearance tracker based on sparse eigentemplate. In *Proc. International Conference on Machine Vision and Applications (MVA2005)*, pages 13–17, 2005.

[11] Y. Sugano and Y. Sato. Person-independent monocular tracking of face and facial actions with multilinear models. In *Proc. IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG2007)*, pages 58–70, 2007.

[12] J. Sung, T. Kanade, and D. Kim. A unified gradient-based approach for combining asm into aam. In *IJCV*, volume 75, pages 297–309, 2007.

[13] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. In *PAMI*, volume 20, pages 39–51, 1998.

[14] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[15] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[16] L. Wiskott, J. M. Fellous, N. Krüger, and C. Malsburg. Face recognition by elastic bunch graph matching. In *PAMI*, volume 19, pages 775–779, 1997.

[17] O. Yamaguchi and K. Furuki. ”smartface”-a robust face recognition system under varying facial pose and expression. *IEICE transactions on information and systems*, E-86-D(1):37–44, 2003.