Engineering

Industrial & Management Engineering fields

Okayama University *Year* 1998

Integration of eigentemplate and structure matching for automatic facial feature detection

Takeshi Shakunaga
Okayama University

Keisuke Ogawa
Okayama University

Shohei Oki
Okayama University

Integration of Eigentemplate and Structure Matching for Automatic Facial Feature Detection

Takeshi Shakunaga   Keisuke Ogawa   Shohei Oki
Okayama University
Department of Information Technology
Tsushima-naka 3-1-1, Okayama-shi, Japan
shaku@it.okayama-u.ac.jp

Abstract

An algorithm is proposed for facial feature detection from a facial image. The algorithm consists of the bottom-up and the top-down interpretation processes, which work with the feature matching module and the structure matching module. Experimental results show that the proposed algorithm can detect no less than five features in 99.3 % of the frontal views as well as it can work even if the face orientation is unknown.

1. Introduction

Facial feature detection is very important for human recognition[1], [4], [2] as well as facial expression recognition and other applications for realizing friendly human interfaces [5]. Various strategies have been proposed for the feature detection for 25 years. Among them the eigentemplate approach, proposed by Turk and Pentland [3], has been extended to cover wide varieties of view-based face recognition in the framework of probabilistic visual learning [6], [4]. This paper shows an extension of the eigentemplate approach for the structure analysis of the face.

2. Facial feature detection system

2.1. System overview

The feature detection task is defined as the estimation of the positions of facial features from a facial image. Both a view-based matching and a structure matching are utilized for the task. The view-based matching is accomplished in an eigenspace which covers all the target features. The structure matching is accomplished with a standard 3d face model and pose estimation from the feature correspondences.

Figure 1 shows main data flow in the system. The detection system comprises the bottom-up and the top-down interpretation processes. In the bottom-up process, feature candidates are enumerated from the input image by the feature matching module. After pruning inconsistent combinations of feature candidates, the structure matching module evaluates and compares possible combinations using the backprojection of the standard 3d face model in the estimated poses. Consequently the bottom-up process can obtain some feasible feature combinations with the estimated pose and feature positions.

After the bottom-up process, the top-down process selects the optimal combination by detecting missing features of all the feature combinations. The missing features are iteratively searched in the neighborhood of backprojected positions with the refined pose of the optimal combination unless no new features are detected.

2.2. Face features and 3d face model

Eight facial features are used for the feature detection. The features consist of the left and right pairs of eyes, eyebrows and ears, as well as the tip of the nose and the center of the mouth, as shown in Fig. 2.

A standard 3d model, as shown in Fig. 3, is used for the structure matching. The face model includes 3d positions of the eight facial features of which images are shown in Fig. 2. The 3d pose of the face is specified by three pose parameters, ψ, θ and ϕ, which denote rotations around x-, y- and z-axes, respectively. Three scale parameters, s_x, s_y and s_z, are used to cover the variations in scales along the three axes. We assume that $s_z : s_x$ is constant in this paper, because we don't use any depth information at all.

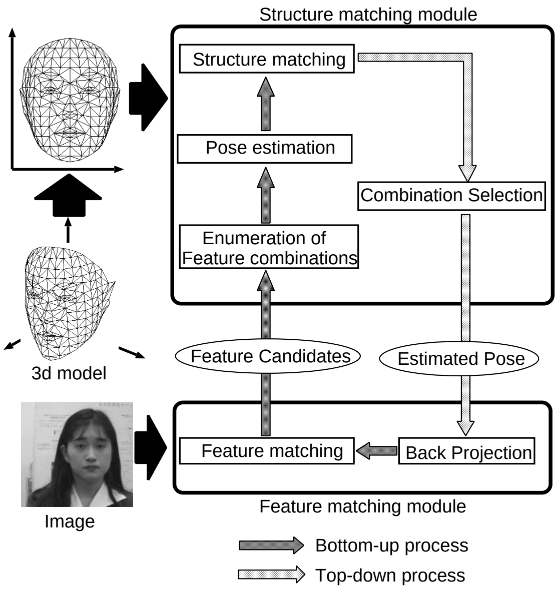


Figure 1. Overview of the feature detection.

2.3. Two matching modules

The feature matching module enumerates feature candidates from a given image by the template matching in an eigenspace. The eigenspace is constructed from learning sets of eight face features in the frontal, left and right face views. The details are shown in 3.

In the bottom-up process, the structure matching module enumerates feasible combinations of feature candidates and finds the 10 most consistent combinations out of them. Then the module estimates the face pose and feature locations using a standard 3d model as well as 2d covariances of feature locations. The detail algorithm is shown in 4.

Once the bottom-up process finds the feasible combinations, the top-down process selects the best one by backprojection, and refines it as well as feature locations using the both modules, as described in 5.

3. Feature matching

3.1. Construction of common eigenspace

A common eigenspace is used in the feature matching module. The eigenspace is constructed from learning sets of the eight features in the frontal, left and right face views. The common template size, with 32 pixels in height and 64 pixels in width, is used with all the features so as to construct the common eigenspace.

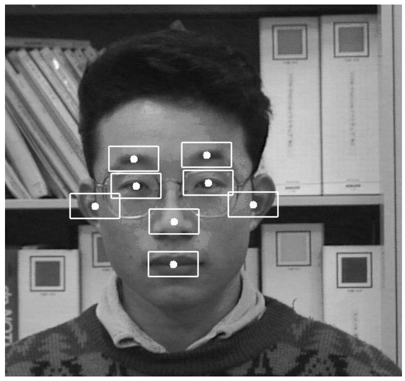


Figure 2. Eight facial features.

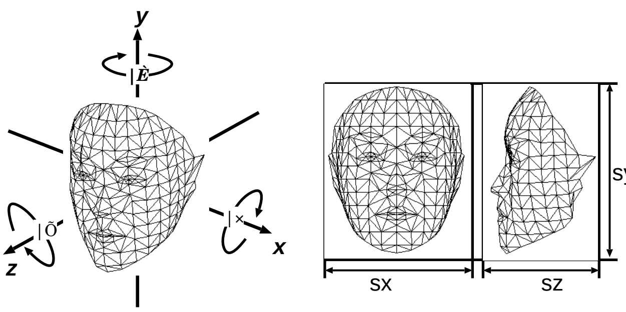


Figure 3. Standard 3d face model.

To accomplish the eigenspace construction, it is necessary to adjust manually each feature in each image I_d^k. For this purpose, a canonical position $\mathbf{u}_{fd}^1 = (u_{fd}^1, v_{fd}^1)$ is marked by a human operator in the canonical image I_d^1 of the d-th direction. The position of the f-th feature in the k-th image $\mathbf{u}_{fd}^k = (u_{fd}^k, v_{fd}^k)$, is adjusted to the canonical position $\mathbf{u}_{fd}^1$ by automatic pattern matching in the neighborhood of the point given by the human operator.

Given a training set of 32-by-64 templates for the eight features, a form training set of vectors $\mathbf{x}_i$s, where $\mathbf{x} \in R^{2048}$. Solving the eigenvalue problem

$$\Lambda = \Phi^T \Sigma \Phi$$

where Σ is the covariance matrix, Φ is the eigenvector matrix of Σ, and Λ is the corresponding diagonal matrix of eigenvalues. Using m eigenvectors which correspond to the m largest eigenvalues, a principal component feature vector $\mathbf{y} = \Phi_m^T \tilde{\mathbf{x}}$ is obtained, where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ is the mean-normalized image vector and Φ_m is a submatrix of Φ containing the m principal eigenvectors.

3.2 Learning in the eigenspace

Let I_{fd}^k denote the k-th training image of the f-th feature in the d-th direction and $k = 1, 2, ..., K$. For the f-th feature in the d-th direction, the mean vector $\mathbf{y}_{fd} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}_{fd}^k$ where $\mathbf{y}_{fd}^k = \Phi_m^T\tilde{\mathbf{x}}_{fd}^k$, and the covariance matrix Σ_{fd} are registered in the dictionary. The maximum DFFS values are also recorded for the effective feature detection:

$$DFFS_{fd}^* = \max_k \ ||\tilde{\mathbf{x}}_{fd}^k - \Phi_m\mathbf{y}_{fd}^k||.$$

3.3 Feature detection

Given a face image, all the feature candidates are detected for all the feature-and-direction pair. The candidate detection is accomplished with the distance-in-feature-space (DIFS) [4], which is calculated by the following Mahalanobis distance in the eigenspace:

$$DIFS_{fdc} = \mathbf{y}_{fdc}{}^T\Sigma_{fd}^{-1}\mathbf{y}_{fdc}.$$

For the efficient candidate detection, images are decoded in a pyramidal structure. Corresponding to the image structure, feature dictionaries are also mapped in each layer of the pyramidal structure. Coarse-to-fine search can be effectively done on the hierarchical structures.

For all the detected candidate $\mathbf{y}_{fdc}(c = 1, ..., 10)$, distance-from-feature-space (DFFS) [4] is calculated by

$$DFFS_{fdc} = ||\tilde{\mathbf{x}} - \Phi_m\mathbf{y}_{fdc}||.$$

If the DFFS distance is less than the threshold $(1.2 * DFFS_{fd}^*)$, the image position $\mathbf{u}_{fdc} = (u_{fdc}, v_{fdc})$ and the dissimilarity D_{fdc} is recorded. Otherwise, the candidate is removed from the candidate list. The pruning by the DFFS is very effective to decrease the calculation cost. In our experiments, only a few candidates can remain after the DFFS check for each fd-pair.

4. Structure matching

4.1 Enumeration of feature combinations

In the structure matching module, all the feasible combinations are enumerated from the feature candidate set.

Let $C_i = \{fdc\}$ denote the i-th feasible combination, where all the face direction d is common in the combination and all the feature f's are different from each other in the combination. Thus all the feasible combinations consist of up to 8 feature candidates in the same direction d. The third suffix c shows the candidate number but it will be omitted from now on for the simple description. Thus, let $C_i = \{fd\}$ denote the i-th feasible combination.

In the enumeration process, geometric consistencies are checked as well as the symbolic enumeration of combinations. That is, inconsistent combinations are not listed up if the sorted order of the elements in u- and v-coordinates are incompatible with that of the dictionary. For example, if the u-coordinates of left and right eyes are in the opposite order, the combination is not considered so far. A lot of combinations can be easily pruned by this check, and only feasible combinations are enumerated for the following process.

4.2 Rough pose estimation

Combining the standard 3d model and each feasible combination of feature candidates, we can make a rough pose estimation. Three pose parameters, θ, ϕ, and ψ are estimated from feature positions. When 4 or more feature points are detected, the pose estimation is formalized as follows:

Pose estimation problem is reduced to an estimation of u_0, v_0 and c_{ij} from a set of (x_{fd}, y_{fd}, z_{fd}) and (u_{fd}, v_{fd}) under orthographic projection.

$$\begin{bmatrix} u_{fd} \\ v_{fd} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix} \begin{bmatrix} x_{fd} \\ y_{fd} \\ z_{fd} \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad (1)$$

where

$$\begin{aligned} c_{11} &= r_x(\cos\theta\cos\phi - \sin\theta\sin\psi\sin\phi) \\ c_{12} &= r_x(\cos\theta\sin\phi + \sin\theta\sin\psi\cos\phi) \\ c_{13} &= -r_x(\sin\theta\cos\psi) \\ c_{21} &= -r_y(\cos\psi\sin\phi) \\ c_{22} &= r_y(\cos\psi\cos\phi) \\ c_{23} &= r_y(\sin\psi) \end{aligned}$$

This problem can be decomposed into the following two (overly constrained) linear simultaneous equations.

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & z_n \end{bmatrix} \begin{bmatrix} u_0 \\ c_{11} \\ c_{12} \\ c_{13} \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & z_n \end{bmatrix} \begin{bmatrix} v_0 \\ c_{21} \\ c_{22} \\ c_{23} \end{bmatrix} \quad (3)$$

The scale parameter r_x, r_y and three pose parameters are estimated from them. When the estimate of θ is far from the direction d, the feature combination is pruned because of the wrong estimate. Note that the sign of ϕ cannot be determined under the orthographic projection. The sign of θ can be determined if only one of the two ears is included in the feature combinations. The sign is not important for our discussion. The sign of ϕ is set to be positive for the following step. If the sign of θ is unknown, it is set to be compatible with the direction d. Thus the transformation matrix can be estimated when 4 or more features are included in a feature combination.

When the number of detected features is less than 4, default values are set to some of the pose parameters and the other parameters are estimated in the similar way. If the number of features is 3, ψ and ϕ are assumed to be 0. If the number is 2, ψ and ϕ are assumed to be 0 and θ is set to be equal to the direction d.

4.3. Learning covariance matrices

In the learning phase, a set of feature positions $\{\mathbf{u}^k_{fd}\}$ can be detected in the training image $I^k_{fd}(k = 1, 2, \ldots K)$, as mentioned in 3.1. Locational covariance matrix Π_{fd} is calculated from $\{\mathbf{u}^k_{fd}\}$ by

$$\Pi_{fd} = \frac{1}{7K}\sum_{g\neq f}\sum_{k=1}^{K}\tilde{\mathbf{u}}^k_{f/g,d}(\tilde{\mathbf{u}}^k_{f/g,d})^T$$

$$\text{where}\quad \tilde{\mathbf{u}}^k_{f/g,d} = \mathbf{u}^k_{fd} - \mathbf{u}^k_{gd} - \frac{1}{K}\sum_{h=1}^{K}(\mathbf{u}^h_{fd} - \mathbf{u}^h_{gd}).$$

4.4. Evaluation of feature combination

A structural dissimilarity is defined for the feature combination $C_i = \{fd\}$ as follows:

$$D1(C_i) = \sum_{fd\in C_i}(\mathbf{u}_{fd}-\mathbf{u}^*_{fd})^T\Pi^{-1}_{f\theta}(\mathbf{u}_{fd}-\mathbf{u}^*_{fd})+\lambda(8-|C_i|)$$

where $\mathbf{u}^*_{fd}$ is an estimated position of the feature by using the pose estimated in 4.2, $\Pi_{f\theta}$ is calculated by the interpolation from $\{\Pi_{fd}\}$, $|C_i|$ is the number of features in C_i and λ is a constant penalty for missing features.

The first term of $D1(C_i)$ shows a sum of normalized dissimilarities to the member features while the second term is a penalty for the non-member features. Because the first term is normalized with a variation of Mahalanobis distance, the penalty constant λ is set to be 9.

On the other hand, a total feature dissimilarity is defined for $C_i = \{fd\}$ as follows:

$$D2(C_i) = \sum_{fd\in C_i} DIFS_{fd} = \sum_{fd\in C_i}\mathbf{y}_{fd}{}^T\Sigma^{-1}_{fd}\mathbf{y}_{fd}.$$

We use a sum of structure and feature dissimilarities, $D(C_i) = D1(C_i) + D2(C_i)$, as an evaluation function for the feature combination. Thus the 10 optimal combinations $\{C^*_i\}$ are detected by comparing $D(C_i)$.

5. Top-down feature detection

In the first stage of the top-down process, the missing features are searched in the neighborhood of backprojected positions of the features for each combination C^*_i. This search updates the C^*_i, its pose parameters as well as the value of evaluation function $D(C^*_i)$. After the first stage, only one candidate C^* is selected out of $\{C^*_i\}$, and the others are pruned.

Once the optimal feature combination C^* is detected, the missing features are iteratively searched in the neighborhood of backprojected positions with the refined pose unless no new features are detected.

In the top-down process, only the feature templates in the nearest direction are used from those of the three directions. The top-down detection is also accomplished with the DIFS in the eigenspace.

6. Experimental results

6.1. Input images and dictionaries

Data specification is summarized in Table 1. Facial images were taken from a fixed camera in the laboratory under the natural lighting condition. The 200 persons, looking forwards, were sitting on the rotation chair in the fixed distance from the camera. The chair was rotated to get the three (frontal, left and right) images for each person. Figure 4 shows sample images of four persons in the three directions. We call the 200 images in the same direction, the frontal set, the left set, and the right set, respectively.

The following two dictionaries are made up for the experimental comparisons.

Dict1 20 faces are used in each direction in the learning phase

Dict2 50 faces are used in each direction in the learning phase

Note that all the dictionary images are excluded from the test images for Dict1 and Dict2.

Table 1. Data specification

# of persons	200
male : female	134 : 66
with/without glasses	48 : 152
directions (0: frontal)	0, 30(left), -30(right)
image size	512 × 480 pixels

Figure 4. Examples of face images

6.2. Definition of correct detection

Correct feature positions are provided in all the images by a human operator for the quantitative evaluation of the detection. Some features in the images are not provided by the human operator, because they are occluded by something. These features are noted and not counted in the statistics.

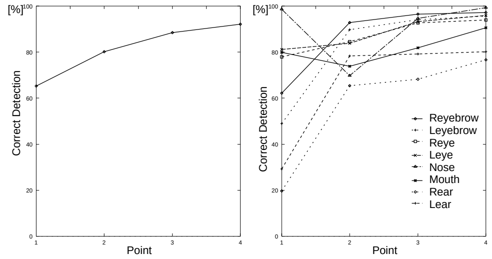
After the experiments, the detection rate is made up by discriminating the correctly detected features and the others. If a feature is detected within a distance threshold D_{max} from the correct position, the feature is said to be correctly detected. Otherwise, the feature is said to be misdetected.

The detection rates are checked at the three points in the process of feature detection as well as the final result.

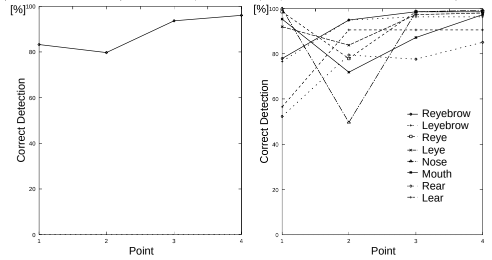
[Point 1] Just after the bottom-up feature detection.
[Point 2] Just after the bottom-up structure matching.
[Point 3] Just after the first top-down feature detection.
[Point 4] Final result.

6.3. Experiment 1 : Frontal set

For the frontal set, Fig. 5 shows average and individual detection rates at Point 1-4. The results with Dict1 are shown in Fig. 5 (a)(b), and those with Dict2 are shown in Fig. 5 (c)(d). The figure shows that Dict2 could correctly detect much more features than Dict1. The detection rate with Dict2 is 96.0% in average. The individual rates are 97-100% with six features except ears. The rates are 85 % and 90 % with right and left ears, respectively.



(a) Average (Dict1) (b) Each feature (Dict1)



(c) Average (Dict2) (d) Each feature (Dict2)

Figure 5. Results for the frontal set.

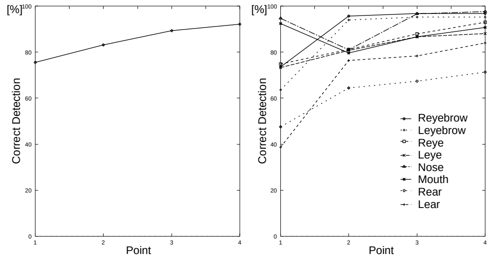
Table 2. Number of detected features for the frontal set.

the number	≤ 2	≤ 3	≤ 4	≤ 5	≤ 6
rate(%)	99.3	99.3	99.3	99.3	97.3

Table 2 shows a distribution of the number of detected features in the results with Dict2. Five or more features were correctly detected in 149 of 150 images in the frontal set, and six or more features were detected in 146 images.

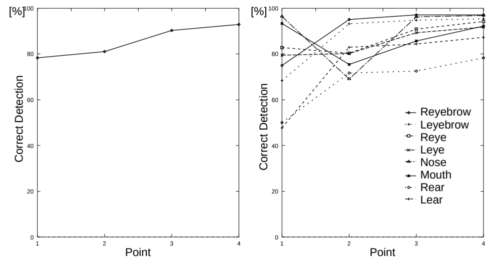
6.4. Experiment 2: Left/right sets

For the left and right sets, Fig. 6 shows average and individual detection rates with Dict2 at Point 1-4. Figure 6(a) shows that the average detection rate is 92 %, and a little worse than that with the frontal set. The individual rates are a little worse than those with the frontal set. The rates with ears are only 84 % for the Right set, and 71 % for the Left set. The intermediate rate with left/right eye at Point 1 is much less than that of the other eye in the Right/Left set. But the final rate with the eye is almost equal to that of the other eye. This improvement shows an effect of the top-down process.



(a) Average (b) Each feature

Figure 6. Results for the left/right sets.



(a) Average (b) Each feature

Figure 7. Results for the whole images.

6.5. Experiment 3: Mixed set

Finally the experiments were accomplished with Dict2 for the whole images. The average detection rates are shown in Fig. 7(a) and individual rates are shown in Fig. 7(b). The final detection rate is 92.7 % in average over three directions. The final rates are 95.8% over the Frontal set, 93.2% over the Right set, and 90.9% over the Left set. They are almost equal to the results in Experiments 1 and 2. The individual rates are 97 % with nose, 92 % with mouth, 92 % with eyes, and 96 % with eyebrows, and 82 % with ears over the whole images. The result examples are shown in Fig. 8.

7. Conclusions

A feature detection algorithm is proposed by integrating feature matching and structure matching. The algorithm consists of the bottom-up and the top-down interpretation processes, which work with the feature matching module and the structure matching module. Experimental results show that the proposed algorithm can detect no less than five features in 99.3% of the frontal views as well as it can work even if the face orientation is unknown. The algorithm seems to be effective for the face identification/verification as well as the facial expression recognition.

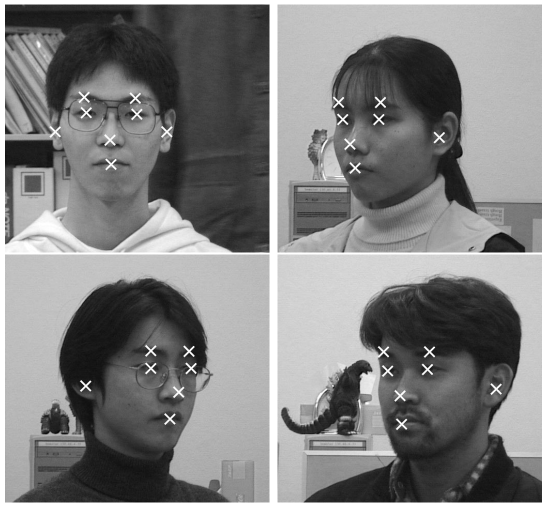


Figure 8. Result examples.

Acknowledgment

This work was partly supported by Research for the Future Program, the Japan Society for the Promotion of Science (Project ID: JSPS-RFTF96P00501).

References

[1] Kanade, T., "Picture Processing by Computer Complex and Recognition of Human Faces," PhD Thesis, Kyoto University, 1973.

[2] Akamatsu, S., Sasaki, T., Fukamachi, H., and Suenaga Y., "Automatic extraction of target images for face identification using the sub-space classification method," IEICE Trans. Inf. & Syst., vol. E76-D, no. 10, pp. 1190-1198, 1993.

[3] Turk, M. and Pentland, A., "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.

[4] Moghaddam, B. and Pentland, A., "Probabilistic Visual Learning for Object Representation", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 696-710, 1997.

[5] Lanitis, A., Taylor, C. J. and Cootes, T. F., "Automatic Interpretation and coding of face images using flexible models," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 743-756, 1997.

[6] Kirby, M and Sirovich, L., "Application of the Karhunen-Loeve Procedure for the characterization of human faces", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1, 1990.