A Graded Approach to Shape Representation

PO-CHI CHEN, WEN-CHANG TSAI, AND SHU-YUEN HWANG

Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, Republic of China

Received August 31, 1993; accepted October 31, 1995

1. INTRODUCTION

Human beings perceive real-world objects in a graded manner, with macroviews providing global information about the objects and microviews providing more detailed information. Consequently, researchers in computer vision have directed their efforts toward building graded representations of objects, especially shapes of objects. Most previous approaches employed operations such as smoothing, blurring, or grouping to gradually obtain representations with different degrees of detail. Parameters of these operations are used to indicate the degree of detail. One objection to using operator parameters is that these parameters are difficult for viewers to interpret intuitively. To overcome this objection, we propose using a graded approach guided by a given similarity requirement to construct a more concise shape description. Our shape representation matches human intuition about detailing degree and suits machine requirement for concise description.

2. OVERVIEW OF THE APPROACH

In this paper, we assume a shape is a closed contour that can be described by a list of vectors. Our general approach is illustrated by the flow chart in Fig. 1. There are four execution blocks in our approach. The first block executes starting-vector detection that is devised to construct an initial condition for computations that follow. In this block, we construct a skeleton for the input shape.
3. PRELIMINARY

In our approach, we use a cluster to denote a part of the input contour which is to be approximated by a token. A token is information which describes part of the shape. One token corresponds to one principle cluster and is mainly devised from that cluster. However, vectors in the neighboring clusters may also affect that token. To consider the effects of neighboring vectors, we introduce the concept of membership degree. We assume that each token is affected by three clusters, one main cluster and two neighboring clusters. This means each token is supported by three clusters of vectors, and each cluster can be said to support three tokens. One token is called the main support token of the cluster and the other two are called neighboring supporting tokens. The membership degree referred to above denotes the degree of support each cluster gives to the three tokens. When a token is formed, we measure how similar it is to its main cluster. This measure is called the token similarity ratio. In our approach, we find a representation that agrees with human requirement on perception similarity. Therefore, the minimum token similarity ratio can be taken as the similarity ratio of the whole set of tokens. For convenience, we will first introduce these concepts: tokens, clusters, membership degree, and similarity ratio.

3.1. Cluster and Token

As mentioned, our approach is performed by clustering the contour vectors, finding the corresponding tokens, calculating the tokens’ attributes, and calculating the similarity ratios until a satisfactory result is obtained. A cluster here is a connected group of neighboring vectors. It can also be viewed as one element in a partition of the input contour. The entire set of clusters is called cluster record.

For convenience, we use $X$ to denote the input shape contour and use $\{x_1, x_2, \ldots\}$ for the vectors in $X$. The notation $N_X$ denotes the number of vectors of $X$. Also, we use $C$ to denote the cluster record and $\{c_1, c_2, \ldots\}$ for the clusters in it. We use $N_C$ to denote the number of clusters in $C$. A similarity ratio remains unchanged as the total number of tokens is reduced.
token usually corresponds to a cluster. We use $t_i$ to denote the token corresponding to $c_i$. We use $T$ to denote the entire set of tokens corresponding to the entire set of clusters $C$. There are three attributes associated with each token: center vector, orientation, and scale. These attributes are defined below.

**Center Vector.** The center vector of a token is a position which can represent the token. We use $\bar{c}_i$ to denote this attribute for token $t_i$; it is calculated by the locations of its formative vectors, as

$$
\bar{c}_i = \frac{\sum_{x_k \in \Omega_i} u(x_k, t_i)^2 x_k}{\sum_{x_k \in \Omega_i} u(x_k, t_i)^2},
$$

where $\Omega_i$ denotes a set of vectors that support the properties of $t_i$, i.e., $\Omega_i = \{c_{i-1} \cup c_i \cup c_{i+1}\}$. The notation $u(x_k, t_i)$ used in the definition above is the membership degree of $x_k$ in the token $t_i$ and is described in next section.

**Orientation.** The second attribute is the orientation, which is denoted by $\theta_i$. The orientation of a token is defined as an optimally fitting direction such that the sum of weighted distance squares is minimized. This sum is denoted by $\Psi_i$ and can be calculated as

$$
\Psi_i(\theta_i) = \sum_{x_k \in \Omega_i} (u(x_k, t_i) d_{ki})^2,
$$

where $d_{ki}$ denotes the distance between $x_k$ and $t_i$. The distance between a vector to a token means a vertical distance between them. Equation (2) needs to be minimized to derive the orientation. Without loss of generality, we assume that $\theta_i$ is $(0, 0)$ and $|\theta_i|$ is 1. Let $\theta$ be the angle between $x_k$ and $\theta_i$, then the following equations will hold.

$$
\cos^2(\theta) = \left(\frac{\langle x_k, \theta_i \rangle}{\|x_k\|}\right)^2 = 1 - \sin^2(\theta),
$$

$$
\sin^2(\theta) = 1 - \left(\frac{\langle x_k, \theta_i \rangle}{\|x_k\|}\right)^2.
$$

$$
d_{ki}^2 = \|x_k\|^2 \sin^2(\theta) = \|x_k\|^2 - (\langle x_k, \theta_i \rangle)^2.
$$

$(x, y)$ denotes the dot product of vectors $x$ and $y$. Equation (2) can then be rewritten as

$$
\Psi_i(\theta_i) = \sum_{x_k \in \Omega_i} (u(x_k, t_i) d_{ki})^2 = \sum_{x_k \in \Omega_i} (u(x_k, t_i)^2 (d_{ki})^2)
$$

$$
= \sum_{x_k \in \Omega_i} u(x_k, t_i)^2 (\|x_k\|^2 - (\langle x_k, \theta_i \rangle)^2)
$$

$$
= \sum_{x_k \in \Omega_i} u(x_k, t_i)^2 \|x_k\|^2 - \sum_{x_k \in \Omega_i} u(x_k, t_i)^2 (\langle x_k, \theta_i \rangle)^2,
$$

where $d_{ki}$ is a straight distance from $x_k$ to $t_i$. Because the membership degrees of vectors are fixed, the first term is constant relative to $\theta_i$. Since $\Psi_i$ is minimized if and only if the second term is maximized, we now try to maximize the second term which can be rewritten as

$$
\sum_{x_k \in \Omega_i} u(x_k, t_i)^2 (\langle x_k, \theta_i \rangle)^2
$$

$$
= \sum_{x_k \in \Omega_i} u(x_k, t_i)^2 \theta_i^T (x_k x_k^T) \theta_i
$$

$$
= \theta_i^T \left\{ \sum_{x_k \in \Omega_i} u(x_k, t_i)^2 x_k x_k^T \right\} \theta_i.
$$

The solution is well known: the matrix $M_i$ is $[\sum_{x_k \in \Omega_i} u(x_k, t_i)^2 x_k x_k^T]$ and $e_i$ is the unit eigenvector of the matrix $M_i$ corresponding to its largest eigenvalue.

**Scale.** The third attribute is the scale which is denoted by $s_i$. The scale we refer to here means the scope of the token. In general, we can use the number of vectors that support the token as the scale value. However, because each vector is weighted, we need to take such weights into consideration. The scale is obtained from the weighted count of the support vectors. We normalize this weighted count by dividing it by the number of total vectors $N_X$, as

$$
s_{i} = \left[ \frac{\sum_{x_k \in \Omega_i} u(x_k, t_i)}{N_X} \right].
$$

### 3.2. Membership Degree

As mentioned, each vector supports three tokens. Therefore, for each vector $x_k$, we define three membership degrees, one for the main token, say $t_i$, and two for the neighboring tokens, say $t_{i-1}$ and $t_{i+1}$. In this paper, we consider three factors concerning the membership degree:

1. ordinal interval between vector $x_k$ and the middle vector of $c_i$,
2. distance between vector $x_k$ and the center of $t_i$, and
3. distance between vector $x_k$ and $t_i$. 

(8)
In the first factor, the cluster $c_i$ is the cluster corresponding to the main token. The middle vector of $c_i$ is the vector whose ordinal number refers to the middle vector in the cluster. This factor reflects the relationship of a vector to the cluster it is in. The second factor is the distance from the center of the token $t_i$ to the vector $x_k$. Unlike the second factor, the third factor refers to the vertical distance between one vector and one token. In considering these three factors, we devised the following formula to represent their total effect to membership degree

$$
\bar{r}_{i,k} = \alpha \|k - \mu_x\|^2 + \beta \|x_k - \bar{c}_i\|^2 + \gamma \|d_{ki}\|^2,
$$

(9)

where the sum of $\alpha$, $\beta$, and $\gamma = 1$ and $\gamma \leq \beta \leq \alpha$.

Member degree can be defined as

$$
u(x_k, t_i) = \left\{ \begin{array}{ll}
\sum_{j}\frac{\bar{r}_{j-i+1,k}}{\bar{r}_{j,k}}^{-1} & \text{if } x_k \in c_{i+1}, \\
\sum_{j}\frac{\bar{r}_{j,k}}{\bar{r}_{j-i+1,k}}^{-1} & \text{if } x_k \in c_i, \\
\sum_{j}\frac{\bar{r}_{j-i+1,k}}{\bar{r}_{j,k}}^{-1} & \text{if } x_k \in c_{i-1}, \\
0 & \text{otherwise.}
\end{array} \right.
$$

(10)

(11)

(12)

where $u(x_k, t_i)$ represents the main support token of $x_k$ is $t_i$. The notations $u(x_k, t_i)^{-1}$ and $u(x_k, t_i)^{+1}$ denote the membership degrees of $x_k$ with respect to neighboring tokens.

To simplify succeeding operations, we define $u(x_k, t_i)$ as

$$
u(x_k, t_i) = \left\{ \begin{array}{ll}
u(x_k, t_i)^{-1} & \text{if } x_k \in c_{i+1}, \\
u(x_k, t_i)^{0} & \text{if } x_k \in c_i, \\
u(x_k, t_i)^{+1} & \text{if } x_k \in c_{i-1}, \\
0 & \text{otherwise.}
\end{array} \right.
$$

(13)

3.3. Similarity Ratio

Two kinds of similarity ratios are used in our approach: one for individual token and one for the entire set of tokens. The token similarity ratio for the token $t_i$ is denoted by $\bar{\sigma}_i$, which can be derived from the distance between $t_i$ and $x_k$. For convenience, we use an average square to reflect an aggregation of the distances between $t_i$ and all vectors in a cluster. This square, denoted by $\bar{d}_i$, is called distance square error and can be defined as

$$
\bar{d}_i = \left[ \frac{\sum_{x_k \in c_i} d_{ki}}{N_{c_i}} \right]^2,
$$

(14)

where $N_{c_i}$ is the number of vectors in $c_i$.

For consistency, we assume that the value range of similarity ratio is between 0 and 1. We also assume that this similar ratio value will decrease as the distance squared error increases. As a result, the similarity ratio is a decreasing function of $\bar{d}_i$, with $\bar{\sigma}_i = 1$ when $\bar{d}_i = 0$, and $\bar{\sigma}_i = 0$ when $\bar{d}_i \to +\infty$. In this paper, we use a sigmoid function that satisfies this requirement as our similarity ratio. Thus we define the similarity ratio of token $t_i$, $\bar{\sigma}_i$, as

$$
\bar{\sigma}_i = \exp[-\lambda(\bar{d}_i)],
$$

(15)

where $\lambda$ represents the equation $10 \times K^2/N_X^2$ and $K$ is a constant. The variable $\lambda$ contains the length of the contour $N_X$. Both the square of $N_X$ and the distance square error $\bar{d}_i$ are influenced by uniform scaling. The influence of uniform scaling in the calculation of similarity ratio is balanced by dividing the distance square error $\bar{d}_i$ by $N_X$. Only spatial attributes are changed by uniform scaling (other attributes are not affected). Therefore the approximation satisfies the requirement of uniform scaling free.

The similarity ratio of the whole set of tokens is the weighted sum of its member tokens. The weight attached to each token is its scale. The whole similarity ratio can be defined as

$$
\bar{\sigma}_T = \sum_{i=1}^{N_T} \bar{s}_i \times \bar{\sigma}_i
$$

(16)

where $\bar{s}_i$ is the scale of the token. This is defined in the following section.

4. THE GRADED APPROACH

This section describes our approach in detail. The algorithm of our main procedure is illustrated in Fig. 3. The input to the procedure is a shape contour $X$ and a stated similarity ratio value $\sigma$. The output is a set of tokens $T$ which approximates $X$.

In step 1, the starting-vector-detection procedure finds the starting vector and the index of the best vector to split. Using the starting vector to represent the contour with a one-cluster cluster record $C$ and the index $q$, later procedures will split this cluster into two clusters for further processing. In step 2, $p$ indicates which cluster will be split in the next pass. Initially of course, there is only one cluster in the cluster record and $p$ is 1.

Steps 3 to 10 form a do loop where the main operations are performed. In step 3, the procedure split splits the $p$th cluster into two at the $q$th vector. The following procedure token detection is the principal step in the loop. This procedure finds an optimal cluster record $C$ and its corresponding set of tokens $T$. The number of the clusters in $C$ does not change during the execution of the procedure. After
Algorithm Graded-approach;

- **Input:** \( \sigma \) and \( X \). /* \( \sigma \) is the required value of similarity ratio and \( X \) is the input shape contour */
- **Output:** \( T \). /* \( T \) is an approximation to \( X \) */

1: \( \text{call starting-vector-detection}(X, C, q); \)
2: \( p \leftarrow 1; \) /* index of the cluster to be split */
3: \( \text{do} \)
4: \( \text{call split}(C, p, q); \) /* split \( p \)th cluster in \( C \) at vector \( x_q \) */
5: \( \text{call token-detection}(X, \varepsilon, C, T); \)
6: \( \forall t_i \in T, \text{calculate } \hat{\sigma}_i; \)
7: \( w \leftarrow j \text{ where } \hat{\sigma}_w = \min_{t_i \in T} \{ \hat{\sigma}_i \} \)
8: \( p \leftarrow w; \)
9: \( q \leftarrow (c_w + s + c_{w+1} + s)/2; \)
10: \( \text{until } \hat{\sigma}_w \geq \sigma; \)
11: \( \text{call merge}(T, C); \)

FIG. 3. The algorithm of our graded approach.

token detection is executed, steps 6 and 7 find the weakest token—the one with the lowest similarity ratio, \( t_w \). The similarity ratio is denoted by \( \sigma_w \). This weakest token and its associated cluster are then indicated for the next iteration loop if its \( \sigma_w \) value is less than the required value. In each case, we increase the number of clusters by one and repeat the loop.

It is obvious that the do loop will terminate. At line 5, we always select the weakest token to split. The weakest token is the token with least similarity to its corresponding curve. If a curve cannot be properly approximated by one line, it must be approximated by two lines. Therefore, the \( \sigma_w \) is continually decreasing and the do loop will terminate.

When the do loop is terminated, i.e., the \( \hat{\sigma}_w \) of each token \( t_i \) is larger than the required value \( \sigma \), a merger procedure is invoked to merge any two adjacent tokens whose orientations are very similar. The goal of merging is to obtain a more human-intuitive approximation. To determine how similar the orientations of two adjacent tokens have to be before they are merged, we design a parameterized formula which takes \( \sigma \) and \( \varepsilon \) into consideration. The parameter \( \varepsilon \) is used to define the degree of improvement that is required. In this formula, \( \rho_1 \) and \( \rho_2 \) are variables and are dependent on the domain of application. Formally, the value of \( Y \) for merge procedure is defined as

\[
Y = \rho_1 \varepsilon + \rho_2 (1 - \sigma),
\]

where \( \rho_1 \) and \( \rho_2 \) are prefixed constants.

4.1. Starting-Vector Detection

The procedure starting-vector detection is illustrated in Fig. 4. In this procedure, we first construct the skeleton.

Procedure starting-vector-detection\((X, C, q)\)

- **Input:** \( X \) /* the shape contour */
- **Output:** \( C \) /* the the initial cluster record and approximation */

0: find the skeleton of \( X \), \( S \);
1: for \( i \leftarrow 1 \) to \( N_S \) do
2: for \( j \leftarrow 1 \) to \( N_S \) do
3: if \( i \neq j \) then
4: Cut the contour by \( s_i \) and \( s_j \) into \( Q_i \) and \( Q_j \);
5: \( R_{ij} \leftarrow |Q_i|/|Q_j|; \)
6: Select \((q_1, q_2)\) such that \( R_{q_1q_2} \) is larger than and most closed to 1;
7: \( N_C \leftarrow 1; \)
8: \( \nu_{c_i} \leftarrow q_1; q \leftarrow q_2; \)

FIG. 4. The procedure for starting-vector detection.
of the input contour $X$. This is done by a one-pass two-operation method first proposed by Arcelli and Baja [1]. The obtained skeleton then intersects the contour with a set of vectors $S$.

Given the set of intersection vectors $S$, we select the starting vector which will best suit the following computations. The for loop (steps 1 to 5) is to check all pairs of contour intersection and calculate the ratios of the lengths of the cut subcontours. In step 6, we select two vectors $s_i$ and $s_j$ whose corresponding ratio is larger than and closest to 1. The following computation begins at the starting vector $s_i$. In a clockwise direction, $Q_i$ is the segment from $s_i$ to $s_j$ and $Q_j$ is the segment from $s_j$ to $s_i$. We select the first vector according to the pair of segments which most equally partition the contour. If $s_{q1}$ and $s_{q2}$ are selected then we let the start index of the cluster $c_1$ be $q_1$ and the index of the vector recommended for splitting be $q_2$.

In Fig. 2, we show an example illustrating the skeleton (the curve in the contour) and its intersection with the contour (black squares).

4.2. Token Detection

The token-detection procedure is illustrated in Fig. 5. This procedure is the kernel of our approach. The input is $X$, $C$, and $\varepsilon$ where $X$ is a shape contour, $C$ is a cluster
record, and $\varepsilon$ is a predefined constant which is used to prevent endless looping. The output includes a cluster record $C'$ and a corresponding set of tokens $T$. Note that the number of clusters in the output cluster record $C'$ is same as that of input cluster record $C$.

In many multiscale line approximation methods, the computation must be taken across lower level approximations, and needs to pursue the redundant coarser scale tokens [4, 14]. However, no efficient pruning technique exists. To solve this problem, we employ a fuzzy concept from Bezdek's fuzzy c-elliptotypes (FCE) algorithm [3] for the membership of a vector $x_k$ to a cluster $c_i$. In our procedure, every vector $x_k$ is assigned to one and only one cluster, say $c_i$. In this case, every vector supports three tokens $t_{i-1}$, $t_i$, and $t_{i+1}$ which correspond to $c_{i-1}$, $c_i$, and $c_{i+1}$, respectively. Each support has a weight which is called its membership and denoted by $u_k^{i-1}$, $u_k^0$, and $u_k^{i+1}$. Initially, let $u_k^{i-1}$ be 0, $u_k^0$ be 1, and $u_k^{i+1}$ be 0 if the vector $x_k$ is in $c_i$. The value may be iteratively modified during the token-detection procedure.

Instead of an optimal result, the token-detection procedure obtains a set of tokens with distortion smaller than a predefined $\varepsilon$. The total distortion $\delta$ can be calculated by the difference between these membership degrees. Three new membership degrees $u_k^{i-1\text{(new)}}$, $u_k^0\text{(new)}$, and $u_k^{i+1\text{(new)}}$ are calculated via Eqs. (10), (11), and (12), respectively. The superscript $\text{(new)}$ is used to distinguish the new degree from the old degree. The function $u^{\text{(new)}}(x_k, t_j)$ is defined similarly to the function $u(x_k, t_j)$, but the values of $u^{\text{(new)}}(x_k, t_j)$ are derived from the new membership degrees. The total distortion $\delta$ is defined by

$$\delta = \sum_{i=1}^{N} \sum_{x_k \in c_i} \sum_{j=i-1, i, i+1} \|u^{\text{(new)}}(x_k, t_j) - u(x_k, t_j)\|. \quad (18)$$

**Reassignment.** When the cluster record has been modified, the new corresponding set of tokens is then found. As mentioned, every vector supports three tokens; when the tokens are changed the membership degree of every vector must be updated accordingly. This update may also
5. EXPERIMENTAL RESULTS

We have applied the approach to several images and obtained sound results. Some are shown in Figs. 7, 8, and 9. In each figure we show a primary shape and its approximation with required similarity ratios of 0.1, 0.3, 0.4, 0.5, 0.7, and 0.9. We found that the desired degree of detailing can be obtained with our simple approach. Using a parameter based on multiple-degree method, we can only give the parameter of the operator but it may not directly reflect the desired degree of detailing. Instead of that, we use a similarity ratio that represents both the desired similarity stated by human viewers and the guiding parameter of the procedure.

6. CONCLUDING REMARKS

In the concluding section, we list several frequently used criteria for shape representation [5, 9, 11] and evaluate our approach according to these criteria.

- Ability to derive general shape properties: Our graded representation can be used to derive general shape properties. For example, we can decompose a shape into several meaningful parts at different degrees of detailing based on the tokens obtained by our approach. We have detailed this decomposition method [13]. The properties of parts such as center, orientation, weight, similarity ratio, protrusion rate, and width contraction rate can be obtained without much extra effort. In Fig. 10, we illustrate an example of such decomposition. We can see that our result shows the symmetric structure of the object, and several parts of the object such as wings and tail rudders. Related information is shown in Table 1.

- Transformation free: Normally, three kinds of transformation are considered, rotation, shift, and scaling. Our method is free from all three kinds. Our starting-vector detection uses the relative positions of intersections of the skeleton with the input shape, therefore rotation of the shape does not affect the result. Shift concerns are minimal since coordinate offsets can easily be removed. Our similarity ratio definition addresses the question of uniform scaling by rendering all attributes of the tokens, except their spatial locations, independent of uniform scaling.

- Uniqueness and invariance: Uniqueness demands that two different objects have different representations. Our approach satisfies this criterion because the given similarity ratio can be as large as one. In this case the representation is the same as input. However, in multitooling cases, uniqueness is not strongly maintained because at lower similarity ratios, different but similar shapes have the same
A GRADED APPROACH TO SHAPE REPRESENTATION

TABLE 1
Part Information of the Decomposed Shape in Fig. 10

<table>
<thead>
<tr>
<th>No</th>
<th>Type</th>
<th>Center</th>
<th>Orientation</th>
<th>σ</th>
<th>Weight</th>
<th>Protrusion</th>
<th>Width</th>
<th>Formative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>(269.392,174.439)</td>
<td>(0.640,-0.768)</td>
<td>0.993</td>
<td>0.123</td>
<td>0.115</td>
<td>0.566</td>
<td>2 − 3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>(226.591,156.247)</td>
<td>(-0.993,0.115)</td>
<td>0.936</td>
<td>0.039</td>
<td>0.007</td>
<td>0.522</td>
<td>3 − 4</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>(237.639,133.886)</td>
<td>(0.840,0.542)</td>
<td>0.967</td>
<td>0.075</td>
<td>0.252</td>
<td>0.643</td>
<td>4 − 5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>(267.761,105.109)</td>
<td>(-0.994,0.169)</td>
<td>0.993</td>
<td>0.139</td>
<td>1.968</td>
<td>0.763</td>
<td>5 − 7</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>(229.525,86.543)</td>
<td>(0.469,0.883)</td>
<td>0.900</td>
<td>0.075</td>
<td>0.070</td>
<td>0.528</td>
<td>7 − 8</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>(201.098,84.134)</td>
<td>(0.018,1.000)</td>
<td>0.935</td>
<td>0.025</td>
<td>0.767</td>
<td>0.495</td>
<td>8 − 9</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>(172.661,86.315)</td>
<td>(-0.488,0.873)</td>
<td>0.986</td>
<td>0.075</td>
<td>0.062</td>
<td>0.527</td>
<td>9 − 10</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>(132.365,102.833)</td>
<td>(0.082,0.187)</td>
<td>0.997</td>
<td>0.131</td>
<td>1.835</td>
<td>0.766</td>
<td>10 − 12</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>(163.490,137.097)</td>
<td>(-0.872,0.490)</td>
<td>0.966</td>
<td>0.078</td>
<td>0.347</td>
<td>0.651</td>
<td>12 − 14</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>(167.536,163.957)</td>
<td>(-0.021,0.571)</td>
<td>0.935</td>
<td>0.030</td>
<td>0.030</td>
<td>0.504</td>
<td>14 − 15</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>(129.379,176.779)</td>
<td>(-0.629,-0.778)</td>
<td>0.989</td>
<td>0.116</td>
<td>0.040</td>
<td>0.546</td>
<td>15 − 16</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>(67.454,200.105)</td>
<td>(0.068,0.250)</td>
<td>0.996</td>
<td>0.209</td>
<td>1.704</td>
<td>0.629</td>
<td>16 − 18</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>(120.494,245.733)</td>
<td>(0.765,0.644)</td>
<td>0.993</td>
<td>0.107</td>
<td>0.017</td>
<td>0.483</td>
<td>18 − 19</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>(162.544,267.903)</td>
<td>(0.686,0.726)</td>
<td>0.944</td>
<td>0.034</td>
<td>0.156</td>
<td>0.468</td>
<td>19 − 20</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>(174.300,283.513)</td>
<td>(-0.791,0.612)</td>
<td>0.960</td>
<td>0.037</td>
<td>0.327</td>
<td>0.504</td>
<td>20 − 21</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>(209.297,325.500)</td>
<td>(0.010,-1.000)</td>
<td>0.962</td>
<td>0.132</td>
<td>1.881</td>
<td>0.895</td>
<td>21 − 27</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>(228.689,283.911)</td>
<td>(0.780,0.625)</td>
<td>0.965</td>
<td>0.039</td>
<td>0.318</td>
<td>0.508</td>
<td>27 − 28</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>(307.266,219.798)</td>
<td>(-0.989,0.149)</td>
<td>0.987</td>
<td>0.248</td>
<td>1.461</td>
<td>0.658</td>
<td>28 − 2</td>
</tr>
</tbody>
</table>

representation. Actually, the uniqueness aspect we concentrated most on was that of making certain shapes with different characteristics have different representations, and based on our discussion of transformation free, we can say that our approach also meets the invariance criterion.

- Stability: The token-detection procedure is a least-squared-error line fitting method whose stability, in computer vision, is considered under noise. Noise is the most difficult issue in computer vision. In our opinion, whether an input is noise or a characteristic should not be determined arbitrarily. Therefore, the problem of noise is resolved in our approach by the similarity ratio which indexes different degrees of abstraction.

- Simplicity: In general, our graded representation is a set of approximations with different degrees of detailing. The complexity of each approximation is different, for example, a smaller similarity ratio requirement corresponds to lower complexity, and a higher similarity ratio to higher complexity. This means different visual tasks can specify different values of the given similarity ratio.

- Ease of implementation: Our approach is conceptually straightforward and simple. In traditional techniques, large numbers of complex differential or integral operators are employed. These mathematical operators will increase the programming and debugging load. Instead of these differential or integral operators, our approach uses only simple operators such as add and average.

- Efficiency: The required computational time and storage capacity of our approach do not satisfy the efficiency criterion at first glance. In our approach, an object is described by a set of approximation representations. In general, four or five approximation representations with different similarity ratios are derived for an object. Our approach

FIG. 10. An example of shape decomposition.
requires more computation time and storage capacity than traditional methods. This is because we are more concerned about later visual tasks, for example, enabling visual recognition to use the lowest initial similarity ratio in obtaining a rough, global classification, and then make the final complex match from among a small number of possibilities. We have developed several applications of the graded approach including the decomposition of a shape to several components [13], and the construction of shape models via learning [6].

**APPENDIX: NOMENCLATURE**

- $\alpha, \beta, \gamma$: parameters used for updating membership degree.
- $C$: a cluster record, i.e., a set of clusters.
- $c_i$: $i$th cluster in cluster record $C$.
- $v_{c_i}$: index of starting vector of cluster $c_i$.
- $\mu_{c_i}$: index of the middle vector of cluster $c_i$.
- $d_{ki}$: the straight distance from $x_k$ to the $t_i$.
- $\delta$: improvement of a reassignment from $t_i$ to $u^{(new)}$.
- $N_S$: the number of the member of set $S$.
- $\Omega_i$: the set of vectors that support the token $t_i$.
- $\tilde{Q}_i$: a partial contour.
- $Y$: the range of closeness for merge procedure.
- $S$: the intersections of the contour and the skeleton.
- $s_i$: the $i$th vector in the set $S$.
- $\sigma$: the required value of the similarity ratio.
- $\bar{\sigma}_T$: the similarity ratio of $T$.
- $T$: an approximation of a shape, i.e., a set of tokens.
- $t_i$: the token that corresponds to $c_i$.
- $\tilde{c}_i$: an attribute, center vector of token $t_i$.
- $\tilde{\sigma}_i$: an attribute, orientation of token $t_i$.
- $s_i$: an attribute, scale of token $t_i$.
- $d_i$: the distance square error of token $t_i$.
- $\bar{\sigma}_i$: the similarity ratio of token $t_i$.
- $u(x_k, t_i)$: the membership degree function of a vector $x_k$ with respect to token $t_i$.
- $\varepsilon$: a given distortion threshold in token-detection procedure.
- $\rho_1, \rho_2$: parameters used in calculating $Y$.
- $X$: the input shape, i.e., a set of vectors.
- $x_k$: the $k$th vector in the set $X$.

**REFERENCES**


PO-CHI CHEN was born on April 21, 1961 in Tainan, Taiwan, Republic of China. He received the B.S. degree in electrical engineering from the National Tsin Hua University in 1984. From 1984 to 1987, he worked at Electronic Research & Service Organization of Industrial Technology Research Institute as a design engineer. In September 1987, he was a graduate student at the Institute of Computer Science and Information Engineering at National Chiao Tung University. In 1989, he received his
M.S. degree. Since 1989, he has been working for his Ph.D. degree. His current research interests include computer vision, machine learning, knowledge discovery in database, and color science.

SHU-YUEN HWANG is a professor in the Department of Computer Science and Information Engineering, National Chiao Tung University. He received B.S. and M.S. degrees in electrical engineering from National Taiwan University in 1981 and 1983, and a Ph.D. degree in computer science from the University of Washington in 1989. He was the director of the Institute of Computer Science and Information Engineering, National Chiao Tung University, from 1993 to 1995. Since 1995, he has served as the deputy director of the Computer Systems Research Center, National Chiao Tung University. His current research interests include artificial intelligence, computer vision, computer simulation, and mobile computing.

WEI-CHANG TSAI was born on March 25, 1979 in Taichung, Taiwan, Republic of China. He received his B.S. and M.S. degrees in computer science and information engineering from the National Chiao Tung University in 1991 and 1993, respectively. Since October 1993, he has been working at the Institute for Information Industry as an assistant engineer. His current research interests include computer vision, machine learning, and computer system integration.