JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 7, No. 2, June, pp. 105–115, 1996
ARTICLE NO. 0010

A Graded Approach to Shape Representation[1]

PO-CHI CHEN, WEI-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

Human beings perceived 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. © 1996 Academic Press, Inc.

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. Good shape representation can present flexible and useful perceptions of objects and also support a wide range of computer vision tasks such as recognition, categorization, reasoning about, and image understanding.

Most previous work used operations such as smoothing, blurring, or grouping to gradually obtain different degrees of detail representation. These efforts included using a number of lines to approximate an original curve [8], using parameters of the smoothing operator [7, 2, 10], using parameters of the symbolic grouping operator [12], and using internal areas and external areas as criteria for multiple information extraction [15]. Parameters of these operations were used to indicate degrees of the information detailing. With respect to computational efficiency, the use of operation parameters might be a good choice for the index of degree of detailing. However, using operation parameters does not directly match human intuition.

To overcome this defect, we devised a graded approach to deriving a shape representation with multiple degrees of detail. Our approach is based on a similarity ratio that measures the similarity between an approximation and an original contour segment. First, a numerical value representing the similarity requirement is stated. Guided by this value, our approach constructs an approximation of the shape that satisfies the stated requirement. In our terminology, an approximation of a shape consists of a set of tokens. A token is a line segment fitted to a partial contour of a shape. Our approach initially uses only two tokens to approximate the given shape. When an optimal result is obtained, we compare the similarity ratio of these tokens to our requirement. If the value obtained is less than our requirement, more tokens are involved to approximate the given shape. This process is iteratively performed until the requirement is achieved. Finally, the most concise representation (least number of tokens) that satisfies the given requirement is generated. The result is that our shape representation matches human intuition about detailing degree and also satisfies the requirement to describe shapes concisely. Our results indicate that our approach yields a more human-intuitive and natural description than others based on operator's parameters. In this paper, we show examples of several shape representations generated by our approach.

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

[1] Research supported by the National Science Council, Republic of China under Grant NSC82-0408-E-009-367.

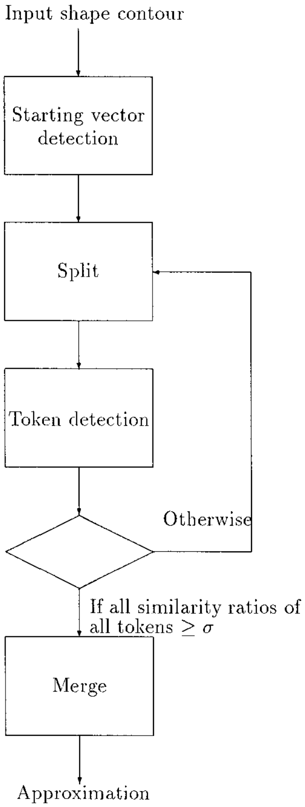
Input shape contour
Starting vector detection
Split
Token detection
Otherwise
If all similarity ratios of all tokens $\geq \sigma$
Merge
Approximation

FIG. 1. Block diagram of the graded approach.

which intersets the shape contour at a set of vectors (Fig. 2). A proper starting vector is selected from the set according to some criterion. This procedure also ensures that our approach is rotation-free.

After starting-vector detection, there is a loop that consists of two blocks. The first block of the loop, *split,* breaks one token in two so that the procedure that follows can devise a finer approximation. The second block in the loop, *token detection* (the kernel of our approach), finds an optimal clustering of the vectors along the shape contour under the constraint that a fixed number of clusters must be found. Each cluster will correspond to a token that contains necessary information about the cluster, and some information about the two neighboring clusters. The similarity ratio between a token and its corresponding cluster of vectors can be computed using their geometrical information. The similarity ratio between the entire approximation and the original shape can then be derived. If the ratio obtained is less than the stated requirement, the loop is executed again. When the requirement is satisfied, the program exits the loop and enters the final block *token merge.*

The purpose of the last block is to obtain a more concise representation while retaining the required quality. This is done by merging neighboring tokens that have close orientations. Because the tokens are so close, the overall similarity ratio remains unchanged as the total number of tokens is reduced.

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 $\{x_1, x_2, ...\}$ 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 $\{c_1, c_2, ...\}$ for the clusters in it. We use N_C to denote the number of clusters in C. A

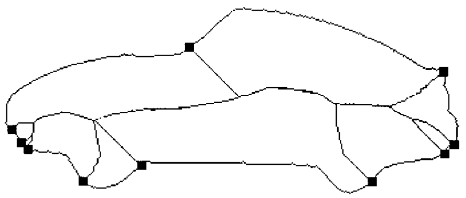

FIG. 2. The skeleton (inside contour) and its intersection with the contour (black squares).

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 $\tilde{c}_{t_i}$ to denote this attribute for token t_i; it is calculated by the locations of its formative vectors, as

$$\tilde{c}_{t_i} = \frac{\sum_{x_k \in \Omega_{t_i}} u(x_k, t_i)^2 x_k}{\sum_{x_k \in \Omega_{t_i}} u(x_k, t_i)^2}, \tag{1}$$

where Ω_{t_i} denotes a set of vectors that support the properties of t_i, i.e., $\Omega_{t_i} = \{c_{i-1} \cup c_i \cup c_{i+1}\}$. 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 $\tilde{o}_{t_i}$. 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 Ψ_i and can be calculated as

$$\Psi_i(\tilde{o}_{t_i}) = \sum_{x_k \in \Omega_i} (u(x_k, t_i) d_{ki})^2, \tag{2}$$

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 $\tilde{c}_{t_i}$ is (0, 0) and $|\tilde{o}_{t_i}|$ is 1. Let θ be the angle between x_k and $\tilde{o}_{t_i}$, then the following equations will hold.

$$\cos^2(\theta) = \left(\frac{\langle x_k, \tilde{o}_{t_i}\rangle}{\|x_k\|}\right)^2 = 1 - \sin^2(\theta). \tag{3}$$

$$\sin^2(\theta) = 1 - \left(\frac{\langle x_k, \tilde{o}_{t_i}\rangle}{\|x_k\|}\right)^2. \tag{4}$$

$$d_{ki}^2 = \|x_k\|^2 \sin^2(\theta) = \|x_k\|^2 - (\langle x_k, \tilde{o}_{t_i}\rangle)^2. \tag{5}$$

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

$$\begin{aligned}
\Psi_i(\tilde{o}_{t_i}) &= \sum_{x_k \in \Omega_i} (u(x_k, t_i) d_{ki})^2 \\
&= \sum_{x_k \in \Omega_{t_i}} (u(x_k, t_i)^2 (d_{ki})^2) \qquad (6) \\
&= \sum_{x_k \in \Omega_{t_i}} u(x_k, t_i)^2 (\|x_k\|^2 - (\langle x_k, \tilde{o}_{t_i}\rangle)^2) \\
&= \sum_{x_k \in \Omega_{t_i}} u(x_k, t_i)^2 \|x_k\|^2 - \sum_{x_k \in \Omega_{t_i}} u(x_k, t_i)^2 (\langle x_k, \tilde{o}_{t_i}\rangle)^2,
\end{aligned}$$

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 $\tilde{o}_{t_i}$. Since Ψ_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

$$\begin{aligned}
&\sum_{x_k \in \Omega_i} u(x_k, t_i)^2 (\langle x_k, \tilde{o}_{t_i}\rangle)^2 \\
&\quad = \sum_{x_k \in \Omega_i} u(x_k, t_i)^2 \tilde{o}_{t_i}^T (x_k x_k^T) \tilde{o}_{t_i} \qquad (7) \\
&\quad = \tilde{o}_{t_i}^T \left\{ \sum_{x_k \in \Omega_{t_i}} u(x_k, t_i)^2 x_k x_k^T \right\} \tilde{o}_{t_i}.
\end{aligned}$$

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

Scale. The third attribute is the scale which is denoted by $\tilde{s}_{t_i}$. 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

$$\tilde{s}_{t_i} = \left[\frac{\sum_{x_i \in \Omega_{t_i}} u(x_k, t_i)}{N_X}\right]. \tag{8}$$

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.

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

$$\hat{r}_{i,k} = \alpha\|k - \mu_{c_i}\|^2 + \beta\|x_k - \tilde{c}_{t_i}\|^2 + \gamma\|d_{ki}\|^2, \tag{9}$$

where the sum of α, β, and γ is 1 and $\gamma \leq \beta \leq \alpha$.

Member degree can be defined as

$$u(x_k, t_i)^{-1} = \left[\sum_{j=i-1,i,i+1} \frac{\hat{r}_{i-1,k}}{\hat{r}_{j,k}}\right]^{-1} \tag{10}$$

$$u(x_k, t_i)^{0} = \left[\sum_{j=i-1,i,i+1} \frac{\hat{r}_{i,k}}{\hat{r}_{j,k}}\right]^{-1} \tag{11}$$

$$u(x_k, t_i)^{+1} = \left[\sum_{j=i-1,i,i+1} \frac{\hat{r}_{i+1,k}}{\hat{r}_{j,k}}\right]^{-1}, \tag{12}$$

where $u(x_k, t_i)^0$ 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

$$u(x_k, t_i) = \begin{cases} u(x_k, t_i)^{-1}, & \text{if } x_k \in c_{i+1}, \\ u(x_k, t_i)^{0}, & \text{if } x_k \in c_i, \\ u(x_k, t_i)^{+1}, & \text{if } x_k \in c_{i-1}, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

3.3. *Similarity Ratio*

Two kinds of similarity ratios are used in our approach: one for individual token and one for th entire set of tokens. The token similarity ratio for the token t_i is denoted by $\tilde{\sigma}_{t_i}$, 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 $\tilde{d}_{t_i}$, is called distance square error and can be defined as

$$\tilde{d}_{t_i} = \left[\frac{\sum_{x_k \in c_i} d_{ki}}{Nc_i}\right]^2, \tag{14}$$

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 $\tilde{d}_{t_i}$, with $\tilde{\sigma}_{t_i} = 1$ when $\tilde{d}_{t_i} = 0$, and $\tilde{\sigma}_{t_i} = 0$ when $\tilde{d}_{t_i} \to +\infty$. 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, $\tilde{\sigma}_{t_i}$, as

$$\tilde{\sigma}_{t_i} = \exp[-\lambda(\tilde{d}_{t_i})], \tag{15}$$

where λ represents the equation $10 \times K^2/N_X^2$ and K is a constant. The variable λ contains the length of the contour N_X. Both the square of N_X and the distance square error $\tilde{d}_{t_i}$ are influenced by uniform scaling. The influence of uniform scaling in the calculation of similarity ratio is balanced by dividing the distance square error $\tilde{d}_{t_i}$ by N_X^2. 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

$$\overline{\sigma}_T = \sum_{i=1}^{N_T} \tilde{s}_{t_i} \times \tilde{\sigma}_{t_i} \tag{16}$$

where $\tilde{s}_{t_i}$ 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 σ. 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 pth cluster into two at the qth 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:   call starting-vector-detection($X, C, q$);
2:   $p \leftarrow 1$; /* index of the cluster to be split */
3:   do
4:       call spit($C, p, q$); /* split $p$th cluster in $C$ at vector $x_q$ */
5:       call token-detection($X, \varepsilon, C, T$);
6:       $\forall t_i \in T$, calculate $\tilde{\sigma}_{t_i}$;
7:       $w \leftarrow j$ where $\tilde{\sigma}_{t_j} = \min_{t_i \in T}\{\tilde{\sigma}_{t_i}\}$
8:       $p \leftarrow w$;
9:       $q \leftarrow (c_w.s + c_{w+1}.s)/2$;
10:  until $\tilde{\sigma}_w \geq \sigma$;
11:  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 similairty ratio, t_w. The similarity ratio is denoted by $\tilde{\sigma}_w$. This weakest token and its associated cluster are then indicated for the next iteration loop if its $\tilde{\sigma}_w$ 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 σ_w is continually decreasing and the do loop will terminate.

When the do loop is terminated, i.e., the $\tilde{\sigma}_{t_i}$ of each token t_i is larger than the required value σ, 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 σ and ε into consideration. The parameter ε is used to define the degree of improvement that is required. In this formula, ρ_1 and ρ_2 are variables and are dependent on the domain of application. Formally, the value of Υ for merge procedure is defined as

$$\Upsilon = \rho_1\varepsilon + \rho_2(1 - \sigma), \tag{17}$$

where ρ_1 and ρ_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 is initial cluster record and approximation. */

find the skeleton of $X$, $S$;
for i$\leftarrow$ 1 to $N_S$ do
    for j$\leftarrow$ 1 to $N_S$ do
        if $i \neq j$ then
            Cut the contour by $s_i$ and $s_j$ into $Q_i$ and $Q_j$;
            $R_{ij} \leftarrow \frac{|Q_i|}{|Q_j|}$;
Select $(q_1, q_2)$ such that $R_{p_1 q_2}$ is larger than and most closed to 1;
$N_C \leftarrow 1$;
$\nu_{c_i} \leftarrow q_1$; $q \leftarrow q_2$;
```

FIG. 4. The procedure for starting-vector detection.

```
Procedure token-detection($X, \varepsilon, C, T$)
•Input: $X, \varepsilon, C$.
•Output: $T$ and updated $C$.

for all $x_k$ do    $u_k^{-1} \leftarrow \emptyset$, $u_k^0 \leftarrow 1$ and $u_k^{+1} \leftarrow \emptyset$;
$\delta \leftarrow 999$; /* initially with a large value */
While $\delta > \varepsilon$ do    /* $\varepsilon$ is a preset value */
  for $i \leftarrow 1$ to $N_C$ do calculate $\tilde{c}_{t_i}$, $\tilde{o}_{t_i}$ and $\tilde{s}_{t_i}$;
  for all $x_k$ do calculate $u_k^{-1\ (new)}$, $u_k^{0\ (new)}$ and $u_k^{+1\ (new)}$;
  calculate $\delta$;    /* $\delta$ is the improvement */
  for all $x_k$ do
    call Reassignment($u_k^{-1\ (new)}$, $u_k^{0\ (new)}$, $u_k^{+1\ (new)}$, $C$, $u_k^{-1}$, $u_k^0$, $u_k^{+1}$);
end While
```

FIG. 5. The token-detection procedure.

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 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 ε where X is a shape contour, C is a cluster

```
Procedure Reassignment ($u_k^{-1\ (new)}$, $u_k^{0\ (new)}$, $u_k^{+1\ (new)}$, $C$, $u_k^{-1}$, $u_k^0$, $u_k^{+1}$);
•Input: $u_k^{-1\ (new)}$, $u_k^{0\ (new)}$, $u_k^{+1\ (new)}$ and cluster-record $C$.
•Output: $u_k^{-1}$, $u_k^0$, $u_k^{+1}$ and cluster-record $C$.

if $u_k^{+1\ (new)} = max(u_k^{-1\ (new)}, u_k^{0\ (new)}, u_k^{+1\ (new)})$
  Reassign $x_k$ to $c_{i+1}$;
  $u_k^{-1} \leftarrow u_k^{0\ (new)}$;
  $u_k^0 \leftarrow u_k^{+1\ (new)}$;
  $u_k^{+1} \leftarrow \emptyset$;
else if $u_k^{-1\ (new)} = max(u_k^{-1\ (new)}, u_k^{0\ (new)}, u_k^{+1\ (new)})$
  Reassign $x_k$ to $c_{i-1}$;
  $u_k^{-1} \leftarrow \emptyset$;
  $u_k^0 \leftarrow u_k^{-1\ (new)}$;
  $u_k^{+1} \leftarrow u_k^{0\ (new)}$;
else
  $u_k^{-1} \leftarrow u_k^{-1\ (new)}$;
  $u_k^0 \leftarrow u_k^{0\ (new)}$;
  $u_k^{+1} \leftarrow u_k^{+1\ (new)}$;
```

FIG. 6. The reassignment procedure.

record, and ε 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^{-1}, u_k^0, and u_k^{+1}. Initially, let u_k^{-1} be 0, u_k^0 be 1, and u_k^{+1} be 0 if the vector x_k is in c_i. The value may be iteratively modified during the token-detection procedure.

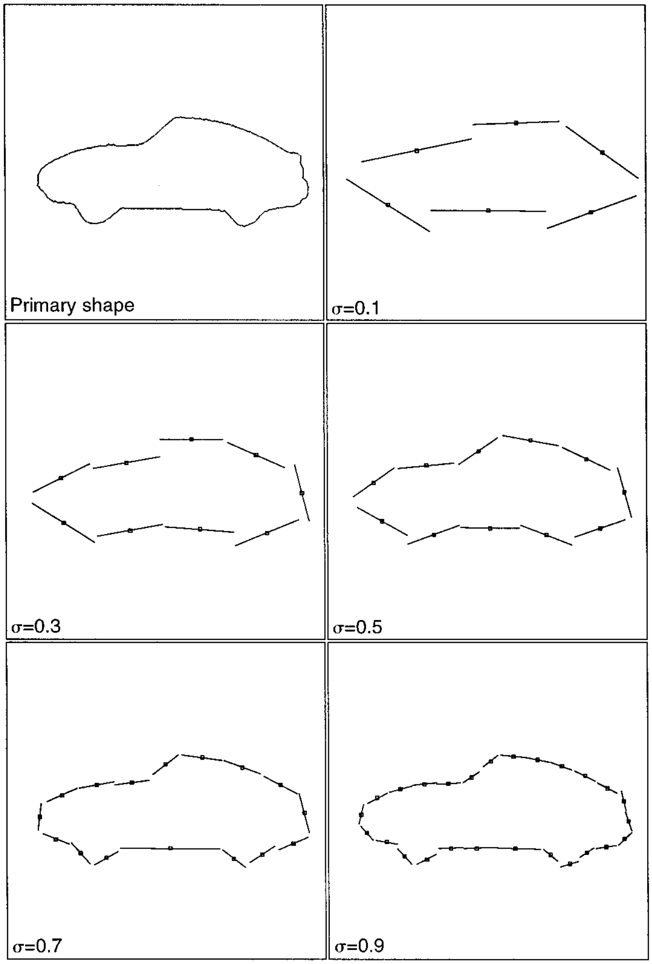

FIG. 7. Car: $\alpha = 0$, $\beta = 0.1$, $\gamma = 0.9$, $K = 70$.

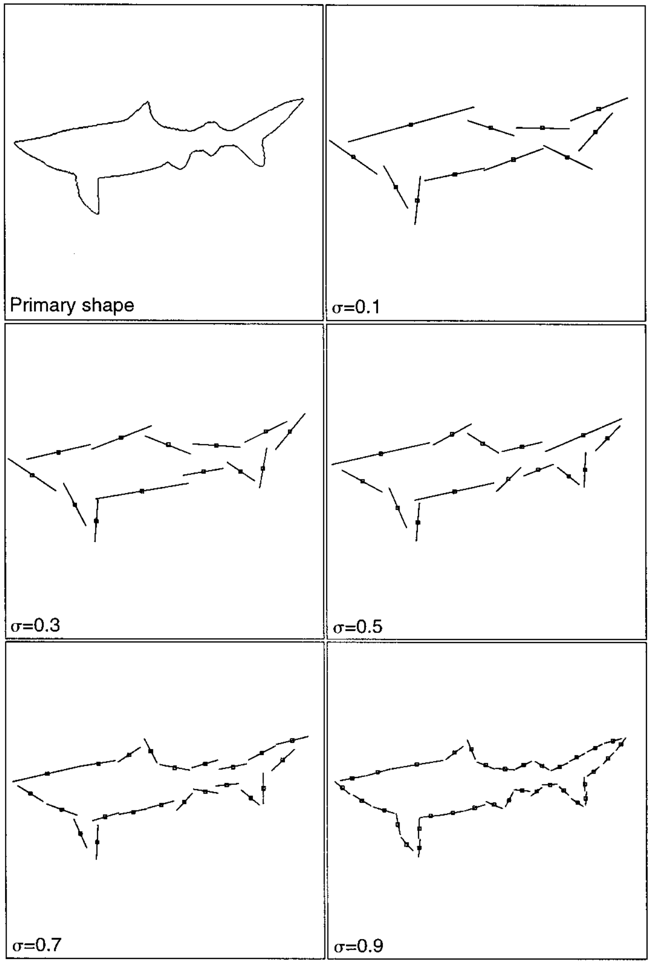

FIG. 8. Fish: $\alpha = 0.032$, $\beta = 0.068$, $\gamma = 0.9$, $K = 90$.

Instead of an optimal result, the token-detection procedure obtains a set of tokens with distortion smaller than a predefined ε. The total distortion δ can be calculated by the difference between these membership degrees. Three new membership degrees $u_k^{-1(\text{new})}$, $u_k^{0(\text{new})}$, and $u_k^{+1(\text{new})}$ 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 δ is defined by

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

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

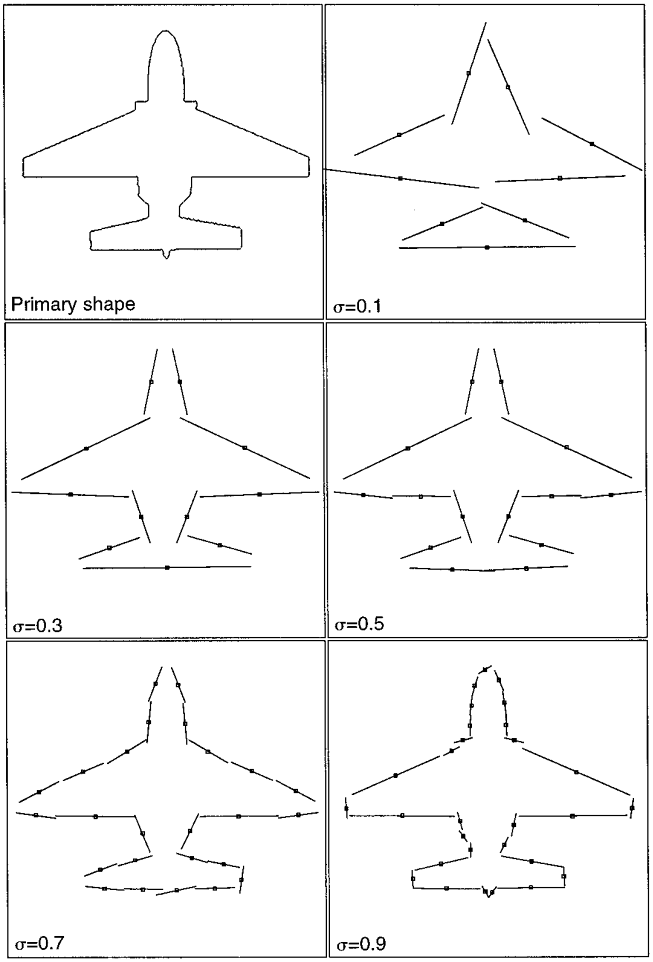

FIG. 9. Plane: $\alpha = 0.02$, $\beta = 0.08$, $\gamma = 0.9$, $K = 110$.

affect the clustering. These operations are performed in the procedure *reassignment.* The algorithm of the reassignment procedure is illustrated in Fig. 6.

A vector supports three neighboring tokens. In other words, a vector possesses three membership degrees for these three tokens. Each vector supports only one main token. Without loss of generality, we assume that the main token a vector x_k supports is t_i. Also, we assume that the corresponding cluster of token t_i is c_i. In the algorithm, we use the subscript k to denote that the membership degree is of vector x_k and token t_i for convenience. The superscript $^{(\mathrm{new})}$ is used to denote that the membership degree is newly obtained. The superscript ^{0} is for the main supporting token and $^{-1}$ and $^{+1}$ are for the neighboring tokens.

This reassignment algorithm is quite straightforward. A vector is assigned to a cluster according to the membership degrees with respect to its main supporting token and those with respect to its two neighboring tokens. If $u_k^{+1(\mathrm{new})}$ is at maximum, then x_k is assigned to c_{i+1}. If $u_k^{-1(\mathrm{new})}$ is at maximum, then x_k is assigned to c_{i-1}. Otherwise, the assignment is not changed.

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 desire 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 multidetailing cases, uniqueness is not strongly maintained because at lower similarity ratios, different but similar shapes have the same

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

No	Type	Center	Orientation	σ	Weight	Protrusion	Width Contraction	Formative tokens
1	1	(269.392,174.439)	(0.640,-0.768)	0.993	0.123	0.115	0.566	2 → 3
2	0	(226.591,156.247)	(-0.993,0.115)	0.936	0.039	0.097	0.522	3 → 4
3	1	(237.639,133.186)	(0.840,0.542)	0.967	0.075	0.232	0.643	4 → 5
4	0	(267.761,105.169)	(-0.994,0.109)	0.993	0.139	1.968	0.783	5 → 7
5	1	(229.525,86.543)	(0.469,-0.883)	0.990	0.075	0.070	0.528	7 → 8
6	0	(201.098,84.134)	(0.018,1.000)	0.935	0.025	0.767	0.495	8 → 9
7	1	(172.661,86.315)	(-0.488,-0.873)	0.986	0.075	0.062	0.527	9 → 10
8	0	(132.365,102.833)	(0.982,0.187)	0.997	0.131	1.835	0.766	10 → 12
9	1	(163.490,137.057)	(-0.872,0.490)	0.986	0.078	0.347	0.651	12 → 14
10	0	(167.536,163.957)	(0.821,0.571)	0.935	0.030	0.030	0.504	14 → 15
11	1	(129.379,176.779)	(-0.629,-0.778)	0.989	0.116	0.080	0.546	15 → 16
12	0	(67.434,200.105)	(0.968,0.250)	0.996	0.209	1.704	0.629	16 → 18
13	1	(120.494,245.733)	(0.765,0.644)	0.993	0.107	0.017	0.483	18 → 19
14	0	(162.544,267.903)	(0.686,-0.728)	0.944	0.034	0.156	0.468	19 → 20
15	1	(174.300,283.513)	(-0.791,0.612)	0.960	0.037	0.327	0.504	20 → 21
16	0	(200.297,325.500)	(0.010,-1.000)	0.982	0.132	1.881	0.895	21 → 27
17	1	(228.689,283.911)	(0.780,0.625)	0.965	0.039	0.318	0.508	27 → 28
18	0	(307.266,219.798)	(-0.989,0.149)	0.987	0.248	1.461	0.658	28 → 2

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 charactersitic should not be determined arbitrarily. Therefore, the problem of noise is resolved in our approach by the similairty ratio which indexes different degrees of abstraction.

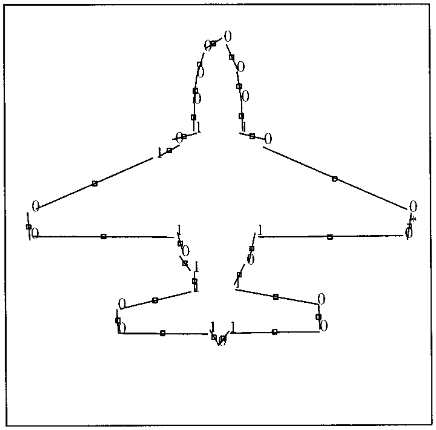

FIG. 10. An example of shape decomposition.

• 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 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

α, β, γ	parameters used for updating membership degree.
C	a cluster record, i.e., a set of clusters.
c_i	ith cluster in cluster record C.
ν_{c_i}	index of starting vector of cluster c_i.
μ_{c_i}	index of the middle vector of cluster c_i.
d_{ki}	the straight distance from x_k to the t_i.
δ	improvement of a reassignment from u to $u^{(\text{new})}$.
N_S	the number of the member of set S.
Ω_{t_i}	the set of vectors that support the token t_i.
Q_i	a partial contour.
Υ	the range of closeness for merge procedure.
S	the intersections of the contour and the skeleton.
s_i	the ith vector in the set S.
σ	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.
$\tilde{c}_{t_i}$	an attribute, center vector of token t_i.
$\tilde{o}_{t_i}$	an attribute, orientation of token t_i.
$\tilde{s}_{t_i}$	an attribute, scale of token t_i.
$\tilde{d}_{t_i}$	the distance square error of token t_i.
$\tilde{\sigma}_{t_i}$	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.
ε	a given distortion threshold in token-detection procedure.
ρ_1, ρ_2	parameters used in calculating Υ.
X	the input shape, i.e., a set of vectors.
x_k	the kth vector in the set X.

REFERENCES

1. C. Arcelli and G. S. D. Baja, A one-pass two-operation process to detect the skeletal pixcels on the 4-distance transform, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(4), 1989, 411–414.
2. A. Bengtsson and J.-O. Eklundh, Shape representation by multiscale contour approximation, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(1), 1991, 85–93.
3. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithm.* A Division of Plenum Publishing Corporation, New York, 1981.
4. M. Boldt and R. Weiss, Token-based extraction of straight lines, in *COINS Technical Report,* pp. 87–104. University Massachusetts, Amherst, 1984.
5. C. C. Chang, S. M. Hwang, and D. J. Buehrer, A shape recognition scheme based on relative distances of feature points from the centroid, *Pattern Recognition,* **24**(11), 1991, 1053–1063.
6. P. C. Chen, W. C. Tsai, Y. N. Yu, and S. Y. Hwang, A learning system for shape representation, in *Proceeding of 1992 IPPR Conference on Computer Vision, Graphics and Image Processing, 1992,* pp. 227–230.
7. G. Dudek and J. K. Tsotsos, Recognizing planar curves using curvature-tuned smoothing, in *Proceedings of 10th ICPR, 1990,* pp. 130–135.
8. M. K. Leung and Y.-H. Yang, Dynamic strip algorithm in curve fitting, *Comput. Vision Graphics Image Process* **51,** 1990, 146–165.
9. J. Ma, C. K. Wu, and X. R. Lu, A fast shape descriptor, *Comput. Vision Graph. Image Process.* **34**(3), 1986, 282–291.
10. F. Mokhtarian and A. K. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(8), 1992, 789–804.
11. F. Mokhtarian and A. K. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(8), 1992, 789–804.
12. Eric Saund, Symbolic construction of a 2-D scale-space image, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(8), 1990, 817–830.
13. W. C. Tsai, P. C. Chen, and S. Y. Hwang, Part decomposition of token-based shape representation, in *Proceedings of 1993 IPPR Conference on Computer Vision, Graphics and Image Processing, 1993,* pp. 228–235.
14. R. Weiss and M. Boldt, Geometric grouping applied to straight lines, in *IEEE Conf. CVPR,* 1986, pp. 489–495.
15. J.-S. Wu and J.-J. Leou, New polygonal approximation schemes for object shape representation, *Pattern Recognition* **26**(4), 1993, 471–484.

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.

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.

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.