Shape Matching Using both Internal and External Morphological Shape Components

Jianning Xu
Computer Science Department, Rowan University,
Glassboro, NJ 08028, USA

Abstract - *A number of morphological shape representation algorithms have been proposed over the years. However, not many shape matching algorithms have been developed based on these representation algorithms. In this paper, we present a structural shape matching algorithm that uses both internal and external shape components selected from the maximal disks determined by a traditional and a generalized morphological skeleton transforms. The algorithm uses relaxation labeling to maintain structural consistency and to derive matching scores. The experiments show that the matching algorithm produces good matching results and it performs better than an earlier algorithm that uses internal components only.*

Keywords: mathematical morphology, shape matching, shape representation, structural matching, relaxation labeling

1 Introduction

Mathematical morphology is a shape-based approach to image processing. It is only natural that a number of morphological shape representation algorithms have been proposed [1-8]. Many of these algorithms use the structural approach. That is, a given shape is described in terms of its simpler shape components and the relationships among the components.

The morphological skeleton transform (MST) is a leading morphological shape representation algorithm [1]. In the MST, a given shape is represented as a union of all maximal disks contained in the shape. The advantages of the MST include that it has a simple and intuitive mathematical characterization as well as an easy and efficient implementation. However, one problem with the MST is that there is a great deal of overlapping among the maximal disks. Some shape matching algorithms have been developed based on the MST [9, 10], but they were only demonstrated to work on some simple geometric shapes.

In a recent paper [11], we described a procedure for selecting structural shape components from the collection of maximal disks determined by a skeleton transform. We also introduced a matching algorithm that uses relaxation labeling to maintain and measure structural consistency when mapping the components of one shape to the components of another shape.

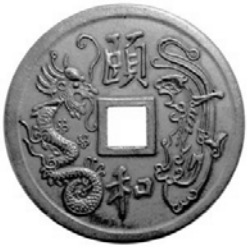

Fig.1. A Chinese coin.

When people describe a shape, not only do they describe the parts in the shape, but they also describe the parts that are not in the shape. For example, the Chinese coin shape in Fig. 1 is probably best described as a circular shape with a square hole in the middle. In this paper, we describe a structural shape representation scheme in which a shape is described using both its internal parts and the parts that are not in the shape. We also present a matching algorithm that uses relaxation labeling to map both internal and external components from one shape to another. Some preliminary experimental results are also reported.

2 Selection of shape components

For the purpose of matching two shapes, we represent a shape as a collection of both internal and external disk components and their relationships. An internal component of a shape represents an area of the shape while an external component represents an area outside the shape. The internal disk components are determined using the standard morphological skeleton transform. A generalized skeleton transform is introduced to determine external shape components.

We first review the standard skeleton transform. For a shape image A and a structuring element B, which is used as the unit disk, we have

$$A = S_N \oplus NB \cup S_{N-1} \oplus (N-1)B \cup \ldots \cup S_2 \oplus 2B \cup S_1 \oplus B \cup S_0 \quad (1)$$

where

$$S_i = (A \ominus iB) \setminus ((A \ominus iB) \circ B) \qquad (2)$$

and N is the largest integer such that $A \ominus NB \neq \emptyset$. Each S_i is called a skeleton subset. Each skeleton point in S_i represents a maximal disk of size i contained in A.

For the unit disk B, the reflection of B is defined as

$$B^r = \{b | -b \in B\} \qquad (3)$$

We now introduce the concept of external skeleton points. For a shape image A and the unit disk B, since closing operation is extensive, we have

$$A = (A \bullet B^r) \setminus T_0 = ((A \oplus B^r) \ominus B^r) \setminus T_0 = (A_1 \ominus B^r) \setminus T_0 \qquad (4)$$

where

$$T_0 = (A \bullet B^r) \setminus A \qquad (5)$$

and

$$A_1 = A \oplus B^r. \qquad (6)$$

Now we can also write

$$A_1 = (A_1 \bullet B^r) \setminus T_1 \qquad (7)$$

where

$$T_1 = (A_1 \bullet B^r) \setminus A_1 = ((A \oplus B^r) \bullet B^r) \setminus (A \oplus B^r). \qquad (8)$$

Combining (4) with (7), we have

$$\begin{aligned} A &= (((A_1 \bullet B^r) \setminus T_1) \ominus B^r) \setminus T_0 \\ &= ((A_1 \bullet B^r) \ominus B^r) \setminus (T_1 \oplus B) \setminus T_0 \\ &= ((((A \oplus B^r) \oplus B^r) \ominus B^r) \ominus B^r) \setminus (T_1 \oplus B) \setminus T_0 \\ &= (A \bullet 2B^r) \setminus (T_1 \oplus B) \setminus T_0. \qquad (9) \end{aligned}$$

Each point in T_0 represents a point that is not in A and each point p in T_1 represents a disk $p \oplus B$ that is not in A. Removing all such points and disks results in the restoration of A from $A \bullet 2B^r$. Following similar steps, we have

$$\begin{aligned} A = (A \bullet nB^r) &\setminus (T_{n\text{-}1} \oplus (n\text{-}1)B) \setminus (T_{n\text{-}2} \oplus (n\text{-}2)B) \setminus \ldots \\ &\setminus (T_1 \oplus B) \setminus T_0 \qquad (10) \end{aligned}$$

where

$$T_i = ((A \oplus iB^r) \bullet B^r) \setminus (A \oplus iB^r). \qquad (11)$$

The points in $T_{n\text{-}1}$, $T_{n\text{-}2}$, … T_1, T_0 can be viewed as external skeleton points and they represents external disks of different sizes. Removal of such disks results in the restoration of A from $A \bullet nB^r$. In fact, these external disks are maximal disks contained in the background shape, or the complement of A. This transform can be viewed as an external skeleton transform.

In our implementation of the skeleton transforms, we use eight structuring elements shown in Fig. 2, instead of a single structuring element, to define discrete disks of different sizes to avoid the generation of non-unit width skeleton segments. In our implementation, the size zero disk $0B$ is defined as $\{(0, 0)\}$. The unit disk B is defined as B_0 and size two disk $2B$ is defined as $B_0 \oplus B_1$. In general, the size i disk iB is defined as

$$iB = (i - 1)B \oplus B_{(i\text{-}1) \bmod 8},\ i > 0. \qquad (12)$$

With this new definition, the set of internal skeleton points of size i is given by

$$S_i = (A \ominus iB) \setminus ((A \ominus iB) \circ B_{i \bmod 8}). \qquad (13)$$

Similarly the new iB^r are defined in terms of $B_0{}^r$, $B_1{}^r$, … $B_7{}^r$. The formula in (11) now becomes

$$T_i = ((A \oplus iB^r) \bullet B^r{}_{i \bmod 8}) \setminus (A \oplus iB^r). \qquad (14)$$

```
---  ---  -+-  ---  ---  --+  ---  +--
-*+  -*-  -*-  -*-  +*-  -*-  -*-  -*-
---  +--  ---  --+  ---  ---  -+-  ---
```

$B_0 \quad B_1 \quad B_2 \quad B_3 \quad B_4 \quad B_5 \quad B_6 \quad B_7$

Fig. 2. Eight two-point structuring elements.

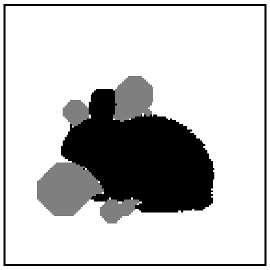

Fig. 3. A rabbit shape and its external components.

Not all the maximal disks will be used in the shape matching. We only use a small number of internal and external disk components to represent a shape. To select the internal components, we first create a list of all the internal skeleton points. The first internal component C_1 is the first disk of size N found in the list. The second component C_2 is the first disk in the list that covers the most area not covered by C_1. In general, component C_m is the first disk in the list that covers the most area not covered by C_1, C_2, … , $C_{m\text{-}1}$. Let M be the total number of internal disk components selected. In our experiments, we set M to 60. Actually, not all these M components will be used in the matching algorithm.

The selection of external components is similar. In our implementation, we set n to 50 in (10). We first create a list of all the external skeleton points. The first external component D_1 is the first disk that covers the most area in $A \bullet 50B^r$. The second component D_2 is the first disk in the list that covers the most area in $A \bullet 50B^r$ not covered by D_1. In general, component D_m is the first disk in the list that covers the most area in $A \bullet 50B^r$ not covered by D_1, D_2, … , $D_{m\text{-}1}$. In our implementation, we select no more than 20 external components. Sometimes, we are not able to select 20 components that all cover non-zero new areas. Fig. 3 shows a rabbit shape and its 20 external components.

We now define structural distances among the internal components. The structural distance from an internal component to itself is defined to be 0. The structural distance between two internal components C_i and C_j is 1 if the two components either overlap or connect to each other. We use the notation $d_s(C_i, C_j) = 1$. The structural distance between C_i and C_j is n (≥ 2), or $d_s(C_i, C_j) = n$, if $d_s(C_i, C_j)$ is not less than n and there exists an internal component C_k such that $d_s(C_i, C_k) + d_s(C_j, C_k) = n$. The structural distances among the external components are defined similarly. Only external components are used to define such distances. The structural distance between two components E_i and E_j of different types is 1 if the two components connect to each other. The structural distance between E_i and E_j is n (≥ 2), or $d_s(E_i, E_j) = n$, if $d_s(E_i, E_j)$ is not less than n and there exists a component E_k of either type such that $d_s(E_i, E_k) + d_s(E_j, E_k) = n$. Note that structural distances among the components of the same type are used to calculate the structural distances between the components of different types.

The actual number of internal disk components used directly in the matching algorithm can be less than M. In our experiments, we use first 30 of the M components. The key features of a shape can typically be captured using 30 internal disk components. The reason for using more internal disk components to calculate internal structural distances is that the results are more accurate. Connected components can become disconnected when not all maximal disks are used to calculate structural distances. However, it seems unnecessary to calculate structural distances for all the internal maximal disks. External components play a lesser role in the matching algorithm. Therefore, no additional external disks are used to calculate structural distances.

Each of the 30 internal components has a size determined by the skeleton transform, but we use the square root of the area of each component to represent its size. For external components, we use the square root of the area of the intersection between a component and $A \bullet 50B^r$ to represent its size. The size normalization is performed by dividing each size number by the size of the largest internal component. If a normalized external component size is greater than 1, we convert it to 1. Another relationship that we use in the matching is the geometric distance between any two components. These distances are also normalized so that the maximum distance is 1. Finally, for each component we also calculate the directions of all neighboring components (with structural distances of 5 or less). Each direction represents the angle between the vector from the current component to the other component and the x-axis.

3 Shape matching using relaxation labeling

Assume that we have two shapes

$$S_1 = \{C_{11}, C_{12}, \dots, C_{1n}\} \tag{15}$$

$$S_2 = \{C_{21}, C_{22}, \dots, C_{2m}\} \tag{16}$$

where each C_{1i} is a disk component of S_1 and C_{2k} is a disk component of S_2. For each shape, the first 30 of these components are internal components and the rest of them are external components. In our experiments, we have $n \leq 50$ and $m \leq 50$. We want to map each C_{1i} to a C_{2k}, but we don't have enough information to select a unique C_{2k}. We use $p_i(k)$ to represent our initial confidence in the hypothesis "C_{1i} is mapped to C_{2k}" for $i = 1, 2, \dots, n$ and $k = 1, 2, \dots, m$. A component can only be mapped to a component of the same type. Therefore, $p_i(k) = 0$ for $i = 1, 2, \dots, 30$ and $k = 31, 32, \dots, m$ and for $i = 31, 32, \dots, n$ and $k = 1, 2, \dots, 30$. Other initial confidence values are determined using size similarity:

$$p_i(k) = \left(1 - \frac{|\text{size}(C_{1i}) - \text{size}(C_{2k})|}{\max\,(\text{size}(C_{1i}), \text{size}(C_{2k}))}\right)\Big/ V_i \tag{17}$$

for $i = 1, 2, \dots, 30$ and $k = 1, 2, \dots, 30$ and for $i = 31, 32, \dots, n$ and $k = 31, 32, \dots, m$. The normalizing constant V_i is defined as

$$V_i = \sum_{k=1}^{30} (1 - \frac{|\,\text{size}(C_{1i}) - \text{size}(C_{2k})|}{\max\,(\text{size}(C_{1i}), \text{size}(C_{2k}))}) \tag{18}$$

for $i = 1, 2, \dots, 30$ and

$$V_i = \sum_{k=31}^{m} (1 - \frac{|\,\text{size}(C_{1i}) - \text{size}(C_{2k})|}{\max\,(\text{size}(C_{1i}), \text{size}(C_{2k}))}) \tag{19}$$

for $i = 31, 32, \dots, n$.

The support from component C_{1j} for the mapping of C_{1i} to C_{2k}, where C_{1i} and C_{2k} are of the same type, is defined as

$$\textstyle\sum_{l=1}^{30} r_{ij}(k,l) p_j(l) \tag{20}$$

for $j = 1, 2, \dots, 30$ or

$$\textstyle\sum_{l=31}^{m} r_{ij}(k,l) p_j(l) \tag{21}$$

for $j = 31, 32, \dots, n$. Note that C_{1i} and C_{1j} can be of different types. Two components of different types can support each other. In (20) and (21), $r_{ij}(k, l)$ represents the strength of compatibility between the hypotheses "C_{1i} maps to C_{2k}" and "C_{1j} maps to C_{2l}." Note that C_{1j} and C_{2l} must be of the same type. A strong support is received if high confidence mappings of C_{1j} are compatible with the mapping of C_{1i} to C_{2k}.The compatibility coefficients $r_{ij}(k, l)$ are defined in terms of properties of and relationships among the components involved. We first consider some special cases. When $i \neq j$, we define $r_{ij}(k, k) = 0$. When $i = j$, expressions (20) and (21) become

$$\textstyle\sum_{l=1}^{30} r_{ii}(k,l) p_i(l) \tag{22}$$

and

$$\textstyle\sum_{l=31}^{m} r_{ii}(k,l) p_i(l) \tag{23}$$

This is the support for the mapping of C_{1i} to C_{2k} from C_{1i} itself. We define $r_{ii}(k, l) = 0$ for $k \neq l$. Now expressions (22) and (23) become $r_{ii}(k, k)p_i(k)$. We define $r_{ii}(k, k)$ to be the similarity between the size of C_{1i} and the size of C_{2k}:

$$1 - \frac{|\text{size}(C_{1i}) - \text{size}(C_{2k})|}{\max\big(\text{size}(C_{1i}), \text{size}(C_{2k})\big)} \tag{24}$$

We also define $r_{ij}(k, l) = 0$ if the structural distance between components C_{1i} and C_{1j} is greater than 5 or the structural distance between components C_{2k} and C_{2l} is greater than 5.

In general, when $i \neq j$ and $k \neq l$, each $r_{ij}(k, l)$ in (20) and (21) is a product of four similarity scores:

$$r_{ij}(k, l) = s_1 s_2 s_3 s_4 \tag{25}$$

We now define these scores. Let d_{s1} be the structural distance from C_{1i} to C_{1j} and d_{s2} be the structural distance from C_{2k} to C_{2l}. Let $e = |d_{s1} - d_{s2}|$. When $\min(d_{s1}, d_{s2}) \leq 2$, we have

$$s_1 = \begin{cases} 1 & \text{if } e = 0 \\ 0.75 & \text{if } e = 1 \\ 0.5 & \text{if } e = 2 \\ 0 & \text{if } e \geq 3 \end{cases} \tag{26}$$

When $\min(d_{s1}, d_{s2}) > 2$, we have

$$s_1 = \begin{cases} 1 & \text{if } e = 0 \\ 0.75 & \text{if } e = 1, 2 \\ 0.5 & \text{if } e = 3, 4 \\ 0 & \text{if } e \geq 5 \end{cases} \tag{27}$$

We are less tolerant to mapping errors in structural distances for structurally closer components. Now consider the vector from C_{1i} to C_{1j} and the vector from C_{2k} to C_{2l}. Let α be the directional difference between the two vectors. When $\min(d_{s1}, d_{s2}) \leq 2$, we define

$$s_2 = \begin{cases} 1 - \frac{3\alpha}{\pi} & \text{if } \alpha < \pi/3 \\ 0 & \text{if } \alpha \geq \pi/3 \end{cases} \tag{28}$$

When $\min(d_{s1}, d_{s2}) > 2$, we define

$$s_2 = \begin{cases} 1 - \frac{2\alpha}{\pi} & \text{if } \alpha < \pi/2 \\ 0 & \text{if } \alpha \geq \pi/2 \end{cases} \tag{29}$$

We are less tolerant to directional errors for structurally closer components. Let d_{g1} be the geometric distance from C_{1i} to C_{1j} and d_{g2} be the geometric distance from C_{2k} to C_{2l}. We define

$$s_3 = 1 - \frac{|d_{g1} - d_{g2}|}{\max(d_{g1}, d_{g2})} \tag{30}$$

Finally, let $t_1 = \text{size}(C_{1i}) / \text{size}(C_{1j})$ and $t_2 = \text{size}(C_{2k}) / \text{size}(C_{2l})$. We define

$$s_4 = 1 - \frac{|t_1 - t_2|}{\max(t_1, t_2)} \tag{31}$$

The support for the mapping of C_{1i} to C_{2k} from all the components in S_1 is

$$q_i(k) = \sum_{j=1}^{30} w_{ij} \sum_{l=1}^{30} r_{ij}(k, l) p_j(l) + \sum_{j=31}^{n} w_{ij} \sum_{l=31}^{m} r_{ij}(k, l) p_j(l) \tag{32}$$

for $i = 1, 2, \ldots, 30$ and $k = 1, 2, \ldots, 30$ and for $i = 31, 32, \ldots, n$ and $k = 31, 32, \ldots, m$. This is a weighted sum of individual supports defined in (20) and (21). The weight w_{ij} represents the strength of influence of C_{1j} on C_{1i}. In calculating this support, we only want to consider the contributions of neighboring components with structural distances of less than or equal to 5. Therefore, we have $w_{ij} = 0$ if the structural distance between C_{1j} and C_{1i} is greater than 5. Otherwise, the value of w_{ij} is defined in terms of the component size, $\text{size}(C_{1j})$, and the structural distance $d_s(C_{1i}, C_{1j})$:

$$w_{ij} = \left(\frac{\text{size}(C_{1j})}{d_s(C_{1i}, C_{1j}) + 1} \right) \Big/ W_i \tag{33}$$

The normalizing constant W_i is defined as

$$W_i = \sum_{j \in N_i} \frac{\text{size}(C_{1j})}{d_s(C_{1i}, C_{1j}) + 1} \tag{34}$$

where $N_i = \{j \mid d_s(C_{1i}, C_{1j}) \leq 5\}$ for $i = 1, 2, \ldots, n$. The larger the size of C_{1j} is, the more important the support of C_{1j} is to C_{1i}. The larger the structural distance between C_{1i} and C_{1j} is, the less important the support of C_{1j} is to C_{1i}.

The support values $q_i(k)$ are used to adjust the confidence values $p_i(k)$. Let $p^0_i(k)$ be the current set of confidence values. The updated set of confidence values $p^1_i(k)$ are determined using the following formula:

$$p_i^1(k) = p_i^0(k) q_i(k) \Big/ \sum_{l=1}^{30} p_i^0(l) q_i(l) \tag{35}$$

for $i = 1, 2, \ldots, 30$ and $k = 1, 2, \ldots, 30$ and

$$p_i^1(k) = p_i^0(k) q_i(k) \Big/ \sum_{l=31}^{m} p_i^0(l) q_i(l) \tag{36}$$

for $i = 31, 32, \ldots, n$ and $k = 31, 32, \ldots, m$. According to (35) and (36), stronger supports will lead to higher confidences and weaker supports will lower the confidences. A new set of support values can be calculated from this new set of confidence values. This process of calculating the support values and updating the confidence values is repeated. The confidence and support values will converge. In our experiments, we use 15 iterations.

The value

$$\sum_{k=1}^{30} p_i(k) q_i(k) \tag{37}$$

with $i \leq 30$ can be seen as the average support that an internal component C_{1i} receives for all its mappings from its neighboring components in S_1. It's a weighted sum of individual supports for specific mappings. The value indicates the level of consistency between the mappings of C_{1i} and the mappings of its neighboring components in S_1. The overall or global consistency score for the whole S_1 is defined

$$F = \sum_{i=1}^{30} v_i \sum_{k=1}^{30} p_i(k) q_i(k) \tag{38}$$

The weight v_i indicates the importance of component C_{1i} and is defined as

$$v_i = \text{size}(C_{1i}) \Big/ \sum_{j=1}^{30} \text{size}(C_{1j}) \tag{39}$$

for $i = 1, 2, \ldots, 30$. The larger the size of C_{1i} is, the more important C_{1i} is in the overall mapping from S_1 to S_2. We use F as a measure of how well components of S_1 are mapped to components of S_2. Note that only internal components are used to calculate the global consistency score. Due to the discrete nature of our disk definition, the external components are not as stable under rotation.

To measure how similar the two shapes are to each other, we first map components of S_1 to components of S_2 and obtain one global consistency score. We then map components of S_2 to components of S_1 and obtain another

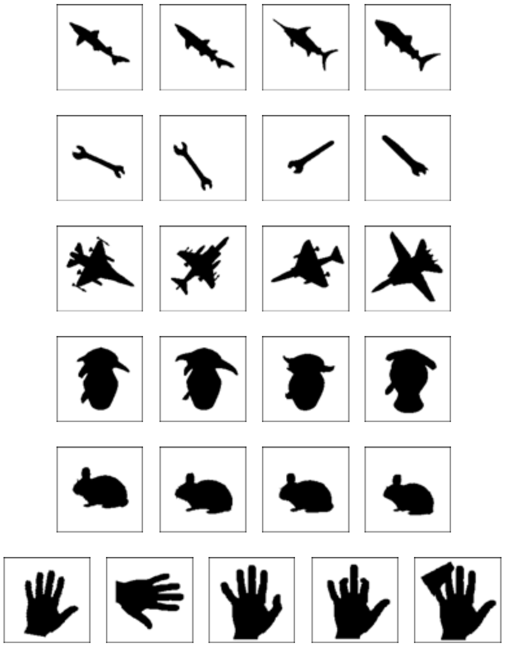

Fig. 4. Kimia data set.

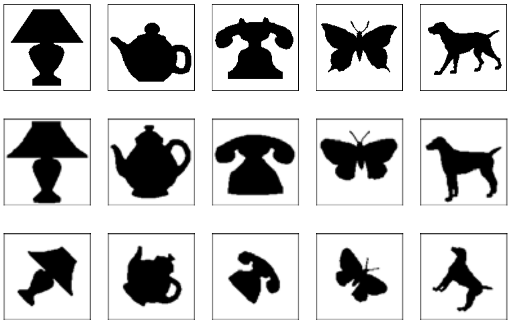

Fig. 5. Shape images used in the experiments.

global consistency score. The average of these two global consistency scores is used as a measure of how similar these two shapes are to each other.

To recognize objects of different orientations, we match S_1 and its 11 rotated versions with rotation angles of $i\pi / 6$, $i = 1, 2, \ldots, 11$, to S_2. The rotation is done to the structural representation of S_1. The best matching score is used to indicate the similarity between S_1 and S_2.

4 Experiments

We performed matching experiments on the Kimia's data set in Fig. 4. This data set contains 25 shapes from 6 categories. Table 1 shows our results in comparison with the results of some existing algorithms [11-15]. In this experiment, we count the number of first, second, and the third closest matches that fall into the same category. Our results are among the best.

Table 1. Matching results on Kimia data set.

Algorithm	Top 1	Top 2	Top 3
Sharvit et al. [12]	23/25	21/25	20/25
Belonie et al. [13]	25/25	24/25	22/25
Ling and Jacobs [14]	25/25	24/25	25/25
Our previous algorithm [11]	25/25	25/25	25/25
Daliri and Torre [15]	25/25	25/25	25/25
Our new algorithm	**25/25**	**25/25**	**25/25**

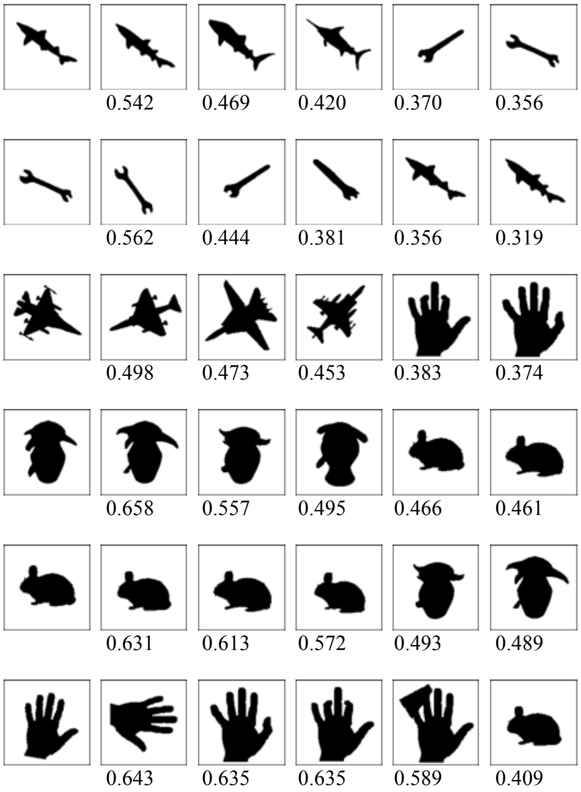

Fig. 6. Five top matches for several shapes.

We also applied our new algorithm and the previous algorithm that uses the internal components only on another set of 15 shapes. The 15 shapes shown in Fig. 5 belong to 5 categories with 3 shapes in each category. In this experiment, we count the number of first and second closest matches that fall into the same category. The algorithm that uses the internal components only produces 15 correct top matches and 14 correct second top matches. Using the new algorithm, we have 15 correct top matches and 15 correct second top matches. The new algorithm performs better than the previous algorithm on this data set. These experiments also show that our algorithms (with and without the external components) can recognize rotated shapes and our algorithms are not very sensitive to scale changes.

In Fig. 6, we also show several of the shapes from Fig. 4 along with their 5 best matches and the corresponding matching scores. For the most part, the results seem to agree well with our intuition. Among shapes from different categories, fish shapes and tool shapes are quite similar. With the big ears, rabbit shapes and animation shapes look quite alike as well.

5 Conclusions

We have developed a structural shape matching algorithm that uses both internal and external shape components selected from the maximal disks determined by a traditional and a generalized morphological skeleton transforms. Only a small number of disk components are used in shape matching. The matching algorithm uses relaxation labeling to optimize the mapping among the components of two shapes. The global consistency scores are used directly as measures of similarity between shapes. Only internal components are used to calculate the global consistency scores. The experiments show that the algorithm produces good and intuitive matching results. The algorithm is tolerant to both rotation and scale changes. This algorithm shows improved performances compared to an earlier morphological shape matching algorithm that uses internal components only.

6 References

[1] P.A. Maragos and R.W. Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, no. 5, pp. 1228-1244, 1986.

[2] I. Pitas and A.N. Venetsanopoulos, "Morphological shape decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 12, no. 1, pp. 38-45, 1990.

[3] P. Maragos, "Morphology-based symbolic image modeling, multi-scale nonlinear smoothing, and pattern spectrum," *Proc. IEEE Comput. Society Conf. Comput. Vision Pattern Recognition*, pp. 766-773, 1988.

[4] J.M. Reinhardt and W.E. Higgins, "Efficient morphological shape representation," *IEEE Transactions on Image Processing*, vol. 5, no. 1, pp. 89-101, 1996.

[5] C. Ronse and B. Macq, "Morphological shape and region description," *Signal Processing*, vol. 25, pp. 91-106, 1991.

[6] J. Goutsias and D. Schonfeld, "Morphological representation of discrete and binary images," *IEEE Transactions on Signal Processing*, vol. 39, no.6, pp. 1369-1379, 1991.

[7] J. Xu, "Efficient morphological shape representation with overlapping disk components," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1346-1356, 2001.

[8] J. Xu, Morphological decomposition of 2-D binary shapes into modestly overlapped octagonal and disk components," *IEEE Transactions on Image Processing,* vol.16, no. 2, pp. 337-348, 2007.

[9] P. E. Trahanias, "Binary shape recognition using the morphological skeleton transform," *Pattern Recognition*, vol. 25, no. 11, pp. 1277-1288, 1992.

[10] E.A. Ei-Kwae and M.R. Kabuka, "Binary object representation and recognition using the Hilbert morphological skeleton transform," *Pattern Recognition*, vol. 33, pp. 1621-1636, 2000.

[11] J. Xu, "Shape matching using morphological shape components and relaxation labeling," *Proc. of 2010 International Conference on Image Processing, Computer Vision, and Pattern Recognition*," Las Vegas, Nevada, July 2010.

[12] D. Sharvit, J. Chan, H. Tek, and B. Kimia, "Symmetry-based indexing of image databases," *J. Visual Comm. And Image Representation*, vol. 9, no. 4, pp. 366-380, 1998.

[13] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 24, pp. 509-522, 2002.

[14] H. Ling and D. Jacobs, "Shape classification using the inner-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286-299, 2007.

[15] M.R. Daliri and V. Torre, "Robust symbolic representation for shape recognition and retrieval," *Pattern recognition*, Vol. 41, no. 5, pp. 1782-1798, 2008.