Shape Matching for Rigid Objects by Aligning Sequences Based on Boundary Change Points

Abdullah N. Arslan[1](✉) and Nikolay M. Sirakov[1,2]

[1] Department of Computer Science and Information Systems,
Texas A&M University – Commerce, Commerce, TX 75428, USA
{Abdullah.Arslan,Nikolay.Sirakov}@tamuc.edu
[2] Department of Mathematics, Texas A&M University – Commerce,
Commerce, TX 75428, USA

Abstract. This paper presents a new boundary (shape) matching algorithm for $2D$ rigid objects without voids. Our new algorithm presents a new shape representation that uses the outcome from an active contour (AC) model. An object's shape is partitioned into a clockwise ordered sequence of *edges*, where every edge is a boundary segment enclosed by *reference points*. These points are convex hull vertices which lie on boundary corners. Further, the reference points are used to generate angles. Hence, a boundary shape maps to a sequence of angles, turning the shape matching problem to alignment of cyclic sequences of angles. The latter makes our method scaling and rotational invariant. Experiments validate the theoretical concept, and provide qualitative comparison with other methods in the field.

Keywords: Shape · Active contour · Concavity · Representation · Cyclic alignment · Matching

1 Introduction

Shape is an effective discriminator for objects in many domains. Shape matching has been used to classify objects in computer vision [13,16,18,28,29,35], in medical imaging [2,24,37], in molecular pharmacology [11,22,23]. This study is part of an on-going project for assessment of threat posed by firearms [4,5].

There are many different shape matching methods based on various representation and distance definitions [9,15,19,25,26,28] (e.g. Haussdorff distance in [19], Inner-Distance in [25]). Some shape matching methods are developed for rigid objects (e.g. [36]) while some others are for non-rigid objects (e.g. [10]). In the present paper, we consider rigid objects without voids.

The method we propose in this paper combines an AC model, and a cyclic sequence alignment method [3,30,32].

A number of methods are published on shape representation and matching using sequences derived from 2D shapes [12,20,21]. Methods in [12,20] are most relevant to our work in this paper. They generate sequences by collecting local

V.E. Brimkov and R.P. Barneva (Eds.): IWCIA 2017, LNCS 10256, pp. 308–321, 2017.
DOI: 10.1007/978-3-319-59108-7_24

information (based on *turning angle*) from the points on the boundary of a given shape. Analogously our method generates sequences from boundary. Unlike [12,20] it considers concavities, convexities, and line segments. The literature has works on concavity extraction, and their use in shape representation and matching [14,31,33].

The basics of the new shape matching method can be found in [3]. Similarly to [3,5] the present paper partitions object boundary into *segments* separated by *reference points* (convex hull vertices which lie on boundary corners). However, unlike previous methods it considers each segment and its enclosing reference points as an *edge*. The shape is described as a sequence of angles constructed from edges. Thus the shape matching consists of sequence matching. The resulting shape matching is more accurate compared to the one in [4,5] which considers reference points and segments separately. The constructed angles do not change under shape rotation and scaling. Therefore, they are invariant according to these plane transformations, and they make the method rotation invariant as well. Furthermore, the method attains scaling invariance in several steps. First, it defines a neighborhood based on the relative sizes of shapes, and removes boundary points in close vicinity. By this way, we aim to have similarly sized boundary representations (in terms of both number of boundary points and lengths of the sequences of angles) for compared shapes. A similarity score is calculated for aligned sequences of angles in two shapes. The total score obtained is divided by the average of the lengths of the compared sequences. This results in scores in $[0, 1]$. The shape similarity score is calculated by using cyclic sequence alignment at a higher level to handle rotations.

We organize sections with the following purposes: Sect. 2 for the notation; Sect. 3 for the new shape representation; Sect. 4 for the new shape matching method; Sect. 5 for experimental evidence; Sect. 6 for concluding remarks.

2 Notations and Basic Definitions

On a given shape without voids, let $B = b_1 b_2 \dots b_n$ be a clockwise ordered sequence of boundary points, $b_j = (x_j, y_j)$ such that $b_n = b_1$. Let $R = r_1 r_2 \dots r_m \subseteq B$ be a clockwise ordered sequence of *reference points*. We call *reference points* of a shape the convex hull points which lie on boundary

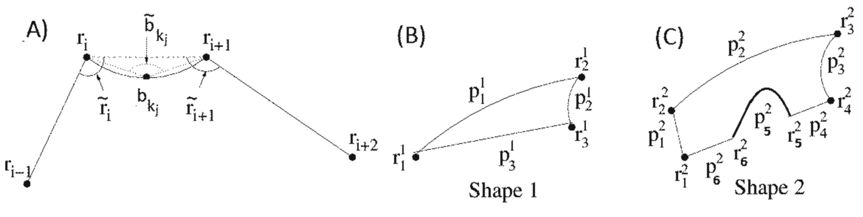

Fig. 1. (A) Edge $e_i = r_i p_i r_{i+1}$; (B, C) Two shapes.

corners, where the convex hull means the convex hull of the shape. For example, in Fig. 1(C), the reference points are $r_1^2, r_2^2, r_3^2, r_4^2, r_5^2$ and r_6^2.

Denote by

p_k: a nonempty boundary *segment* as a sequence $b_{k_1}b_{k_2}\ldots b_{k_{|p_k|}}$, where $b_{k_j} \in B$ denotes the j'th boundary point in segment k;
e_k: an *edge* $r_k p_k r_{k+1}$ which is a segment enclosed by two reference points, i.e. $r_k b_{k_1} b_{k_2} \ldots b_{k_{|p_k|}} r_{k+1}$;
$\tilde{b}_j$: angle $\angle r_k b_j r_{k+1}$ with a vertex at b_j and arms through r_k and r_{k+1};
$\tilde{r}_i$: an angle $\angle r_{i-1} r_i r_{i+1}$ at reference point r_i;
$\tilde{p}_k$: the sequence of angles $\tilde{b}_{k_1}\tilde{b}_{k_2}\ldots\tilde{b}_{k_{|p_k|}}$ at the points in segment p_k;
$\tilde{s}_k$: a sequence of angles $\tilde{r}_k \tilde{p}_k \tilde{r}_{k+1}$.

To summarize, for every segment p_i, there is a corresponding sequence of angles $\tilde{p_i} = \tilde{b}_{i_1}\tilde{b}_{i_2}\ldots\tilde{b}_{i_{|p_i|}}$.

Figure 1(A) illustrates an edge $e_i = r_i p_i r_{i+1}$ that includes a (concave) boundary segment. The angles $\tilde{r}_i = \angle r_{i-1} r_i r_{i+1}$, $\tilde{r}_{i+1} = \angle r_i r_{i+1} r_{i+2}$, and $\tilde{b}_{k_j} = \angle r_i b_{k_j} r_{i+1}$ are also illustrated. Shape 1 in (B) has 3 reference points, and 3 edges that include one convex, one concave, and one line segment. Shape 2 in (C) has 6 reference points, and 6 edges that include 3 line segments, one convex, and two concave segments. No angles are shown in (B) and (C) in Fig. 1, but they are calculated in a similar way as shown in (A).

3 Shape Representation

We abstract a shape by a clockwise ordered sequence of edges e_k obtained from boundary.

3.1 Boundary Extraction

In the present study we apply a shrinking active contour model (S-ACES) to extract the boundary of objects of interest. The model is developed in [32], and uses the following evolution equation to converge S-ACES toward objects' boundaries:

$$r(s,t) = Re^{as-4a^2(t_0+u\partial t)}[\cos(cas), \sin(sas)], \tag{1}$$

In Eq. 1, $s \in [0, \frac{2\pi}{ca}]$ is a space parameter, t is a time parameter, t_0 is initial time moment, $a = |\partial s|/2$, $u = 1, 2, \ldots$, and R is the radius of the initial circle. To make the initial circle encompass the entire image we select:

$$u = 0, a^2 t_0 = 0.001, c = 1000. \tag{2}$$

Denote the image function as $f(x(s,t), y(s,t)) = f(r(s,t))$. The condition (BC) that halts the AC in the vicinity of the object's boundary is:

$$\begin{array}{l} r(s,t) = r(s, t+\partial t)) \text{ which holds if} \\ \frac{\partial f(r(s,t))}{\partial t} > \varepsilon \;\text{ where }\; t = t_0 + u\partial t \;\text{ such that} \\ 2.5 \geq ta^2 \geq 0.001. \end{array} \tag{3}$$

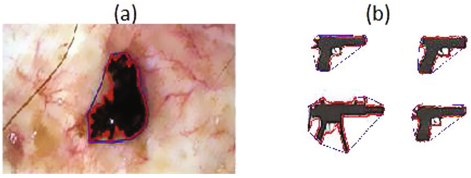

Fig. 2. Extracted CH and boundary of (a) a skin lesion; (b) weapons.

To evolve into deep concavities, a curve re-parametrization is conducted [32]. If Ineq. 4 is satisfied the AC point (x_i, y_i) moves to the right of the AC if a clockwise direction is considered.

$$(y_i - y_{i-1})(x_{i+1} - x_i) < (y_{i+1} - y_i)(x_i - x_{i-1}). \tag{4}$$

The AC was validated on the extraction of 162 skin lesion boundaries and 170 weapon and non-weapon images [5]. Sample results are shown in Fig. 2.

3.2 Generating the Shape Sequence

Let $(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1})$ be any three clockwise ordered points in a 2D Euclidean plane. Consider the clockwise traversal of $(x_{i-1}, y_{i-1}), (x_i, y_i)$, (x_{i+1}, y_{i+1}). If Ineq. 4 is satisfied, we say that (x_i, y_i) is a *concavity point* with respect to (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}). If the reverse of Ineq. 4 is satisfied, then we say that (x_i, y_i) is a convexity point with respect to (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}). If Ineq. 4 becomes an equality, then we say that these points are co-linear.

We define concavity as a sequence of clockwise-ordered boundary points $r_k b_{k_1} b_{k_2} \dots r_{k+1}$ on B such that all b_{k_j} are concavity points with respect to r_k and r_{k+1}. We say that $r_k b_{k_1} b_{k_2} \dots r_{k+1}$ is a *level-1 concavity* if it is a concavity not included in another (larger) concavity (this definition is in parallel with the definition in [31]). In this paper we consider only level-1 concavities. Figure 1(A) illustrates a concave boundary segment.

Our method collects a sample of boundary points on the boundary via S-ACES. We process the initial boundary obtained by S-ACES to eliminate boundary points which are"too close" to each other. For this purpose, we calculate the minimum-area rectangular bounding box enclosing the object. By dividing the perimeter of this rectangle by a parameter, we obtain a threshold length. If any two neighbors are within this threshold (horizontally and vertically) from each other, only one of them is kept in B and the other one is removed. Our goal is to obtain similarly sized sequence representation regardless of the size of the object. We perform a complete clockwise traversal on B starting at an arbitrary point, and find all level-1 concavities. We do this by considering every visited point as a potential concavity beginning, and all successors as potential concavity ends. Then we apply Ineq. 4 to check if all the points between the two potential concavity beginning and end points are concavity points with respect to these points. If the current point is not a concavity beginning we move to the next point. Once a level-1 concavity is found, we mark the found concavity and advance the traversal to the concavity end point, and continue iterating the same logic described above. By this way, we find all level-1 concave segments. All other points on B are labelled as convexity points initially. Additional clockwise traversals are performed to partition them into line, and other (convex) segments using Ineq. 4. Concave and convex segments which are almost linear are replaced by line segments. We merge short line segments into larger ones. This includes consecutive line segments which are almost linear, too. Similarly, two consecutive convex segments can be merged into a larger convex segment. We test if two consecutive segments can be merged by taking the beginning of the first and the end of the second one, and checking if all the points between them have the same characteristic (convex or concave) with respect to these reference points. We continue iterating until there are no such consecutive segments.

We define a *shape* as a cyclic sequence $\tilde{s}_1\tilde{s}_2\tilde{s}_3\ldots\tilde{s}_{|\tilde{s}|}$, where each $\tilde{s}_i$ is a sequence of angles obtained from edge e_i ordered clockwise. In this sequence, each angle at a reference point is calculated by using two other reference points (the predecessor and the successor reference of this point clockwise), and each angle at boundary point is calculated using the reference points enclosing this point. We assign a sign to segment types as follows: for any segment p_i, $sign(p_i) = -1$ indicates that p_i is concave; $sign(p_i) = 0$ indicates that p_i is a line; $sign(p_i) = 1$ indicates that p_i is convex. Figure 1 includes two example shapes. Shape number is shown in the superscript. Shape 1 in (B) is represented by $\tilde{s}^1_1\tilde{s}^1_2\tilde{s}^1_3$, where $\tilde{s}^1_i = \tilde{r}^1_i\ \tilde{p}^1_i\ \tilde{r}^1_{i+1}$, for all $i, 1 \leq i \leq 3$, and $sign(p^1_1) = 1$, $sign(p^1_2) = -1$, $sign(p^1_3) = 0$, and Shape 2 in (C) is represented by $\tilde{s}^2_1\tilde{s}^2_2\tilde{s}^2_3\tilde{s}^2_4\tilde{s}^2_5\tilde{s}^2_6$, where $\tilde{s}^2_i = \tilde{r}^2_i\ \tilde{p}^2_i\ \tilde{r}^2_{i+1}$, for all $i, 1 \leq i \leq 6$, and $sign(p^2_6) = sign(p^2_4) = sign(p^2_1) = 0$, $sign(p^2_2) = 1$, and $sign(p^2_5) = sign(p^2_3) = -1$.

4 Shape Matching

We convert the differences between angles $\tilde{b}^1_i$ and $\tilde{b}^2_j$, and $\tilde{r}^1_k$ and $\tilde{r}^2_\ell$ to similarity scores in $[0, 1]$. All angles are represented in radian and as a factor of π. We convert the difference $\Delta = |\tilde{b}^1_i - \tilde{b}^2_j|$ between angles $\tilde{b}^1_i$ and $\tilde{b}^2_j$, to a similarity score in $[0, 1]$ using

$$f(\Delta) = \begin{cases} 1, & \text{if } (\Delta < \beta_1); \\ 1 - \sqrt{\Delta}, & \text{if } \beta_1 \leq \Delta < \beta_2; \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where in the current implementation we set $\beta_1 = 0.02$, and $\beta_2 = 0.05$. Via these parameters, the differences are either ignored or amplified. The purpose is to distinguish very close matches from other similarities. When the difference Δ is within β_1, the angles are considered perfectly matching, and the similarity score $f(\Delta)$ is maximum (i.e. 1). When Δ is larger than or equal to β_2 the angles are considered completely different (not similar at all), and $f(\Delta)$ is minimum (i.e. 0). In between β_1 and β_2, as the difference Δ increases the similarity score $f(\Delta)$

decreases at a faster rate. We note that with $\beta_1 = 0.02$, and $\beta_2 = 0.05$, $f(\Delta) = 0$ if $\Delta > 0.05$; $f(\Delta) = 1$ if $\Delta \leq 0.02$; and $f(\Delta)$ is in $[0.77, 0.85]$ for Δ in $[0.02, 0.05)$. Analogously, the difference $\Delta = |\tilde{r}^1_k - \tilde{r}^2_\ell|$ between angles $\tilde{r}^1_k$ and $\tilde{r}^2_\ell$, is converted to a similarity score in $[0, 1]$ using $f(\Delta)$. Based on these, the similarity score for a pair of boundary segments p^1_i, and p^2_j, with the same sign, is the alignment score $score_s(p^1_i, p^2_j)$ for the sequences $\tilde{b}^1_{i_1}\tilde{b}^1_{i_2}\ldots\tilde{b}^1_{i_{|p^1_i|}}$, and $\tilde{b}^2_{j_1}\tilde{b}^2_{j_2}\ldots\tilde{b}^2_{j_{|p^2_j|}}$. This can be computed by a special case of the global sequence alignment algorithm [27] in which score of insertions, deletions are zeros, and substitutions (matches) have positive scores. We can also formulate the objective of the optimization as the following:

$$score_s(p^1_i, p^2_j) = \max_{i', j'} \sum_{m=1}^{u} f(|b^1_{i'_m} - b^2_{j'_m}|) \tag{6}$$

over all index sequences i', j' such that $i'_1, i'_2, \ldots, i'_u$ is a subsequence of $i_1, i_2, \ldots, i_{|p^1_i|}$, and $j'_1, j'_2, \ldots, j'_u$ is a subsequence of $j_1, j_2, \ldots, j_{|p^2_j|}$, for some $u \in [1, \min\{|p^1_i|, |p^2_j|\}]$. For p^1_i and p^2_j, when one is concave and the other one is convex then $score_s(p^1_i, p^2_j) = 0$; when one is concave or convex, and the other one is a line segment then the similarity score calculated by using Eq. 6 is halved.

For two sequences of angles $\tilde{s}^1_i$, $\tilde{s}^2_j$ obtained from edges $e^1_i = r^1_i p^1_i r^1_{i+1}$, $e^2_j = r^2_j p^2_j r^2_{j+1}$ with $sign(p^1_i) = sign(p^2_j)$, $wscore_s(\tilde{s}^1_i, \tilde{s}^2_j)$ is one fourth of the sum of the scores between r^1_i and r^2_j, and between r^1_{i+1} and r^2_{j+1} plus half of the alignment score for segments $\tilde{p}^1_i$ and $\tilde{p}^2_j$ divided by average length $(|p^1_i| + |p^2_j|)/2$. The resulting score is in $[0, 1]$ and denoted by $wscore_s$ $(\tilde{s}^1_i, \tilde{s}^2_j)$. More formally,

$$wscore_s(\tilde{s}^1_i, \tilde{s}^2_j) = \frac{1}{4}\left(f(|r^1_i - r^2_j|) + f(|r^1_{i+1} - r^2_{j+1}|)\right) + \frac{score_s(p^1_i, p^2_j)}{|p^1_i| + |p^2_j|} \tag{7}$$

For any two shape sequences $\tilde{s}^1 = \tilde{s}^1_1\tilde{s}^1_2\ldots\tilde{s}^1_{|\tilde{s}^1|}$ and $\tilde{s}^2 = \tilde{s}^2_1\tilde{s}^2_2\ldots\tilde{s}^2_{|\tilde{s}^2|}$ we consider sequence alignment [27] for calculating their similarity score $score_{seq}(\tilde{s}^1, \tilde{s}^2)$. Let $|\tilde{s^1}|$ and $|\tilde{s^2}|$ denote the number of edges in shapes 1, and 2, respectively.

The objective function can be described as the following:

$$score_{seq}(\tilde{s}^1, \tilde{s}^2) = \max_{i,j} \sum_{m=1}^{r} wscore_s(\tilde{s}^1_{i_m}, \tilde{s}^2_{j_m}) \tag{8}$$

over all index sequences i, j such that $i_1, i_2, \ldots, i_r$ is a subsequence of $1, 2, \ldots, |\tilde{s}^1|$, and $j_1, j_2, \ldots, j_r$ is a subsequence of $1, 2, \ldots, |\tilde{s}^2|$, for some $r \in [1, \min\{|\tilde{s}^1|, |\tilde{s}^2|\}]$.

The dynamic programming formulation of the sequence alignment in this case is based on deleting $\tilde{s}^1_i$, inserting $\tilde{s}^2_j$, and matching $\tilde{s}^1_i$ to $\tilde{s}^2_j$. We create a model with the following similarity score parameters described by the real score function γ. We set the insert and delete scores to zero. That is, $\gamma\left(\begin{bmatrix} - \\ \tilde{s}^2_j \end{bmatrix}\right) = \gamma\left(\begin{bmatrix} \tilde{s}^1_i \\ - \end{bmatrix}\right) = 0$. The similarity score for matching $\tilde{s}^1_i$ to $\tilde{s}^2_j$ is

$\gamma\left(\begin{bmatrix}\tilde{s}^1_i\\ \tilde{s}^2_j\end{bmatrix}\right) = wscore_s(\tilde{s}^1_i, \tilde{s}^2_j)$, where $wscore_s(\tilde{s}^1_i, \tilde{s}^2_j)$ is the alignment score calculated as we described in Eq. 7. Then the alignment score $score_{seq}$ $(\tilde{s}^1, \tilde{s}^2) = E_{|\tilde{s}^1|,|\tilde{s}^2|}$, where E is the matrix calculated by the following dynamic programming formula for sequence alignment [27]: For all i, j, $i \in [0, |\tilde{s}^1|]$, $j \in [0, |\tilde{s}^2|]$, $E_{i,-1} = E_{-1,j} = -\infty$, and for all other i, j, if $i = j = 0$ then $E_{0,0} = 0$ else $E_{i,j}$ is calculated from $E_{i,j-1}, E_{i-1,j-1}$, and $E_{i-1,j}$ using the following formula

$$E_{i,j} = \max \left\{ \begin{array}{c} E_{i,j-1} + \gamma\left(\begin{bmatrix}\tilde{s}^1_i\\ \text{-}\end{bmatrix}\right),\ E_{i,j-1}\ + \gamma\left(\begin{bmatrix}\text{-}\\ \tilde{s}^2_j\end{bmatrix}\right), \\ E_{i-1,j-1} + \gamma\left(\begin{bmatrix}\tilde{s}^1_i\\ \tilde{s}^2_j\end{bmatrix}\right) \end{array} \right\} \tag{9}$$

The alignment score $score_{seq}(\tilde{s}^1, \tilde{s}^2) = E_{|\tilde{s}^1|,|\tilde{s}^2|}$ is normalized by dividing it by an upper bound for a maximal attainable score $(|\tilde{s^1}|+|\tilde{s^2}|)/2$, where n_1, n_2 are respectively the number of edges in aligned shapes $\tilde{s}^1$, $\tilde{s}^2$. The resulting score is in $[0, 1]$ and denoted by $|score_{seq}(\tilde{s}^1, \tilde{s}^2)|$. That is, $|score_{seq}(\tilde{s}^1, \tilde{s}^2)| = \frac{score_{seq}(\tilde{s}^1, \tilde{s}^2)}{(|\tilde{s^1}|+|\tilde{s^2}|)/2}$. If $\tilde{s}^1 = \tilde{s}^2$, $|score_{seq}(\tilde{s}^1, \tilde{s}^2)| = 1$, otherwise, $|score_{seq}(\tilde{s}^1, \tilde{s}^2)| < 1$.

Given $\tilde{s}^2$, the *cyclic shift* of $\tilde{s}^2$ by k positions is $\tilde{s}^{2,k} = \tilde{s}^2_{k+1}\tilde{s}^2_{k+2} \ldots \tilde{s}^2_{|\tilde{s}^2|}\tilde{s}^2_1 \ldots \tilde{s}^2_k$. Therefore we define the shape similarity score for two shapes (cyclic sequences) $\tilde{s}^1$ and $\tilde{s}^2$ as

$$cscore(\tilde{s}^1, \tilde{s}^2) = \max_{0 \le k < |\tilde{s}^2|} |score_{seq}(\tilde{s}^1, \tilde{s}^{2,k})| \tag{10}$$

If $\tilde{s}^1 = \tilde{s}^{2,k}$, for some k, then $cscore(\tilde{s}^1, \tilde{s}^2) = 1$, else, $cscore(\tilde{s}^1, \tilde{s}^2) < 1$.

On the bases of above concepts, we develop the following shape matching algorithm: (1) Extract the boundary; (2) Generate shape sequences $\tilde{s}^1 = \tilde{s}^1_1\tilde{s}^1_2 \ldots \tilde{s}^1_{|\tilde{s}^1|}$ and $\tilde{s}^2 = \tilde{s}^2_1\tilde{s}^2_2 \ldots \tilde{s}^2_{|\tilde{s}^2|}$; (3) Shape matching is performed as follows: (3.1) Define T as a two dimensional matrix whose elements are described by the following:

$$T[i, j] = wscore(\tilde{s}^1_i, \tilde{s}^2_j),\ 1 \le i \le |\tilde{s}^1|,\ 1 \le j \le |\tilde{s}^2| \tag{11}$$

The entire matrix T is built by calculating $wscore(\tilde{s}^1_i, \tilde{s}^2_j)$ for all i, j. (3.2) Sequences of angles from edges (Shapes) $\tilde{s}^1$ and $\tilde{s}^2$ are cyclically aligned using matrix T, the optimum score is found by using the dynamic programming formulation given by Eq. 9, and calculating Eq. 10. That is, for $k = 0$ to $|\tilde{s}^2| - 1$, $|score_{seq}(\tilde{s}^1, \tilde{s}^{2,k})|$ is computed, and an optimal alignment is returned.

Boundary extraction takes $O(MN)$ time, where $M \times N$ is the size of the input images. Boundary sequence generation can be done within the same theoretical time complexity using convex hull. In our implementation, we only used the boundary points as described in Sect. 3.2, and it took less than 16 ms to generate boundary sequences. For different k, the scores of the insertions, deletions, and substitutions at given positions in the alignment computation for $\tilde{s}^1$ and $\tilde{s}^{2,k}$ are different because of the cyclic shifting of symbols in $\tilde{s}^2$. Each pair of positions $(i, j + k)$ (i in $\tilde{s}^1$ and j in $\tilde{s}^{2,k}$) in aligning $\tilde{s}^1$ and $\tilde{s}^{2,k}$ corresponds to (i, j) in aligning $\tilde{s}^1$ and $\tilde{s}^2$. Therefore, matrix T is computed only once, and all necessary values are available there at different indices. Each pairwise segment alignment takes time $O(|p_i^1||p_j^2|)$. Let ℓ^1 and ℓ^2 be the number of boundary points of the input shapes, Shape 1 and Shape 2, respectively, after reducing close neighbors as described in Sect. 3.2. That is, $\ell^1 = \Sigma_{k=1}^{|\tilde{s}^1|}|p_k^1|$, and $\ell^2 = \Sigma_{k=1}^{|\tilde{s}^2|}|p_k^2|$. The total time required for building the table T is $O(\Sigma_{i,j}|p_i^1||p_j^2|) = O(|p_1^1|\ell^2 + |p_2^1|\ell^2 + \ldots + |p_{|\tilde{s}^1|}^1|\ell^2) = O((\Sigma_{k=1}^{|\tilde{s}^1|}|p_k^1|)\ell^2) = O(\ell^1\ell^2)$. After constructing the table T, each pairwise edge alignment takes time $O(|\tilde{s}^1||\tilde{s}^2|)$. The algorithm performs $|\tilde{s}^2|$ such alignments. Therefore, the total time spent in this step is $O(|\tilde{s}^1||\tilde{s}^2|^2)$, where $|\tilde{s}^1|$ and $|\tilde{s}^2|$ are the number of edges (much smaller than the perimeters), respectively. Hence the total time of our shape matching algorithm is $O(\ell^1\ell^2 + |\tilde{s}^1||\tilde{s}^2|^2)$.

5 Experimental Results

An earlier version of the shape representation and matching method was used in [5]. The results there validated our theoretical concepts on a visual weapon ontology composed by 153 weapons [5]. A visual hierarchy was designed by creating clusters such as machine guns, pistols, riffles. Figure 3 includes a cluster from this hierarchy. The clustering was done based on the algorithm in [17] and using as the measure of similarity an earlier version of the cyclic shape sequence alignment score described by the present paper. The ontology was queried by objects. The results of identifying queried objects were encouraging [5].

In the present paper we tested our method on the dataset of Aslan and Tari [7], and shown some of our results out of 56 shapes in Fig. 4. In each row we give a query on the leftmost column, and in the next four columns we present the nearest matches to the query in descending order of similarity as computed by our method.

The results in Fig. 4 show the accuracy of 100% of our shape model in finding identical 2D shapes. When segments -in particular concavities- appear similarly in two compared shapes, the similarity score is high. For example, turtle looks similar to human when hands and legs are in similar gesture.

We also want to note an implementation detail. The normalized similarity scores distinguish the nearest neighbors. The scores are numbers in $[0, 1]$. We

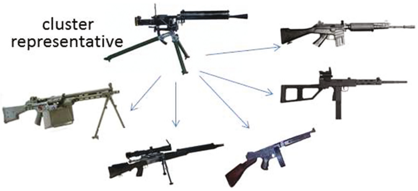

Fig. 3. A cluster from the visual hierarchy in [5].

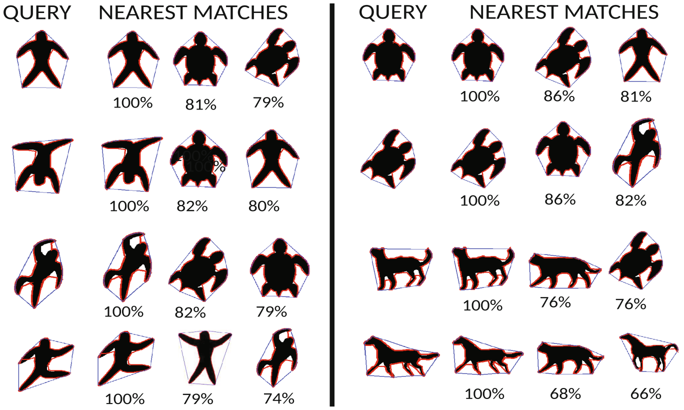

Fig. 4. Select queries and nearest matches in a subset of the database from [7].

note that because of the scoring function for angles in Eq. 5, only very closely matching angles in edges (at reference points, and at boundary points) contribute to the total score; others have no contribution. Our observation is that in this model we discriminate real similarities from other random matches. That is, the scores we obtain is an effective measure of similarity. However, the resulting normalized scores are small, even for near matches. Therefore, by taking the fourth power of the normalized score in $[0, 1]$, we maintain the same ordering for matches, yet we obtain numbers corresponding 70% or higher percentages for near matches. These numbers are shown as percentages in our results.

The dataset of Aslan and Tari [7] contains 56 images. The sizes of these images vary from 190×111 to 222×250 pixels. In [8,34] the same dataset is used for shape classification. The methods in [8,34] took above 5 min to process all images. In our method on this dataset, on average per image, the AC extraction took 400 ms; the sequence generation time, and the total time for alignments were 16 ms. The total time to process the entire data set with our method and answer queries with all 56 images is 47 s using a PC with 1.6 GHz clock, 512 MB RAM. The comparisons show that our method is faster than those in [8,34].

We remark that our shape representation is based on the boundary features (e.g. concavities). This is different than models based on symmetry axis in [7]. Naturally, a symmetry axis-based model performs very well classifying all human shapes with different arm and leg positions. Our shape representation and comparison method performs very well for objects whose boundaries are rigid such as firearms. The effectiveness of the seed ideas and initial method were proven empirically in [5]. The effectiveness in detecting partial matches was illustrated in identifying partially occluded firearms in [4]. We also note that two dissimilar objects can have similar axis of symmetry such as a broom and a long gun, however, boundary features can be the discriminating features in this case. Our method can differentiate these objects in this example from each other (see [5]).

To validate our shape matching method on weapons on a simple illustrative example, we select six weapons from the weapon ontology presented in [5]. For these weapons, the number of boundary points range from 300 to 998, and the boundary sequence lengths range from 97 to 129. These weapons come from three different clusters such that there are two weapons from each cluster. In Fig. 5, on the very left, enclosed by a dashed rectangle these weapons are shown. We perform a query with each weapon. In every case, the weapon itself is the nearest match, and the next nearest match is the other weapon from the same cluster.

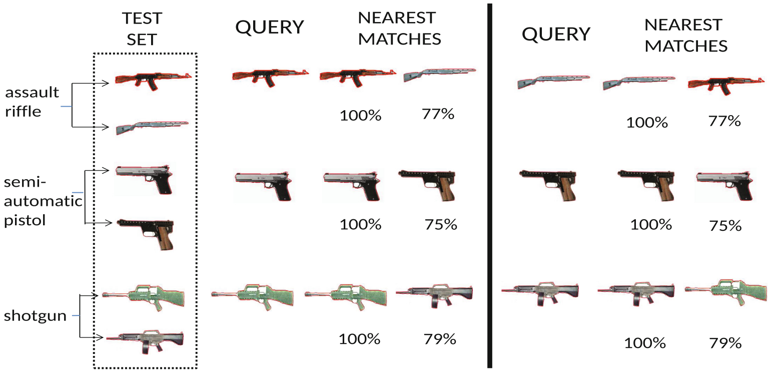

Fig. 5. Select queries and nearest matches in a subset of weapon ontology in [5].

To validate and have an experimental evidence of the scale invariance capability of our method, we compared a query human figure with its 2×2 and 3×3 enlargements shown in Fig. 6(A). The similarity score remains very high even there is a significant scaling difference.

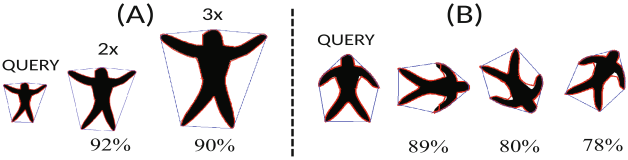

Fig. 6. (A) Comparison of a human figure with its 2× and 3× enlargements; (B) Comparison of a human figure with its rotations.

To validate the rotation invariance capability of our method, we compared a query human figure with its rotations by 30, 90, and 150° angles (clockwise.) We show the results in Fig. 6(B). The similarity score between the object and its rotated version remains high. However, the 90° angle rotation yields better scores compared to 30-degree and 150-degree rotations. We believe the reason lies with the rectangular bounding box. We bound an object with its minimum horizontal, and minimum vertical boundary position as the bottom left corner, and maximum horizontal, and maximum vertical boundary position on the top right corner. To see this take a vertical line segment of length x with width 1; rotate it at 45° angle, the rectangular box that encloses the new object has a diagonal of length $\sqrt{2}\ x$, which has larger perimeter. In 90-degree case, the perimeter stays the same. However, in 30 and 150-degree cases, the perimeter increases. As a result, in these cases, shape sequences are longer, the total length of the compared sequences is larger while the similarity score remains nearly the same, and the normalized score is lower.

Figure 7 includes a clustering result based on the Gonzalez' algorithm [17]. For computing the pairwise distances between shapes, our new algorithm is used. In the figure, instead of distances, the normalized similarity scores are shown as percentages. This example is another validation of our method's performance on clustering/discriminating rigid objects based on their shapes.

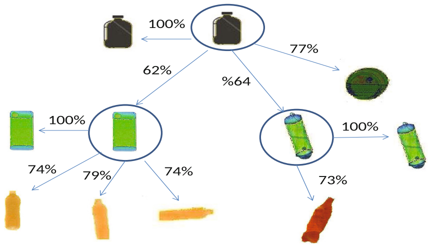

Fig. 7. Clusters of some containers.

6 Conclusions

The paper presents an improved shape representation based on convex, line, and concave edges. These boundary features perform very well as shown in [5] for rigid objects such as weapons which retain these features. This preservation makes them very suitable for detecting partial matches [4]. Further, contributions and advantages of the present study compared to the shape representation and matching approach in [3,5] are the following:

- Here the shape is represented as a single sequence of angles obtained from edges. The previous methods created separate convex hull (CH) and boundary sequences of angles, and aligned them separately and independently. Therefore, the new method is better applicable for local matches when only parts of the boundary are visible. The new method aligns sequences of edges maintaining the original clockwise ordering with respect to each other (in the order of edges). The AC was reformulated for the tasks.

- The new shape matching method generates similarly sized sequences and similar angles even though the object sizes are different, and even though the objects are rotated. This makes the method not only rotation invariant, but also, scale invariant, which is a missing property in [3,5].

One shortcoming of the new representation and method is that some unrelated objects may look similar in different orientation. For example, with respect to the new shape model, a human and a turtle can be very similar, and two human shapes in different posses (e.g. a jumping man in two different poses) may look very different (see. Fig. 4). The new representation and method apply better to rigid objects. This is because rigid objects retain their shapes better and concavities in them could be identifying features.

One disadvantage of the elaborated method is that it uses a number of user-defined parameters such as a number of thresholds.

An area in which our shape matching method can be applied is the Ribonucleic Acid (RNA) 2D structure analysis. RNA molecule makes interesting 2D formations. Similar functions and evolutionary relatedness can be analysed via structural similarities. New representations and algorithms for RNA 2D structures continue to be popular (e.g. see recent articles [1,6]). RNA 2D structures have distinguishable boundary features such as bulges and hairpin loops in their drawings. A linear sequence representation can be developed based on these boundary features, and partitioning the boundary into segments. This would yield a cyclic sequence alignment RNA structure comparison algorithm similar to the one we use in this paper. Such a representation would also be useful for searching boundary for given segment types. Our method will also be applied to compare malignant and benign skin lesion boundaries.

Acknowledgements. We are thankful to reviewers for their insightful and constructive comments. Addressing them yielded valuable additions to the present work. This work is partially supported by USA NSF Award No: IIS-1528027.

References

1. Anandan, J., Fry, E., Monschke, Arslan, A.N.: A fast algorithm for finding largest common substructures in multiple RNAs. In: Proceeedings of the 9th International Conference on Bioinformatics and Computational Biology, BICOB'07, Honolulu, HI, USA, 20–22 March, pp. 51–57 (2017)
2. Antani, S., Xu, X., Long, L.R., Thoma, G.R.: Partial shape matching for CBIR of spine X-ray images. In: SPIE Proceedings of Storage and Retrieval Methods and Applications for Multimedia, vol. 5307, pp. 1–8 (2004)
3. Arslan, A.N., Sirakov, N.M., Attardo, S.: Weapon ontology annotation using boundary describing sequences. In: Proceedings of IEEE SSIAI 2012, pp. 101–104, Santa Fe, 22–24 April 2012
4. Arslan, A.N., Hempelmann, C.F., Attardo, S., Blount, G.P., Sirakova, N.N., Sirakov, N.M.: Identification of partially occluded firearms through partonomy. In: Sadjadi, M. (ed.) Proceedings of SPIE 2015 (2015). doi:10.1117/12.2184102

5. Arslan, A.N., Hempelmann, C.F., Attardo, S., Blount, G.P., Sirakov, N.M.: Threat assessment using visual hierarchy and conceptual firearms ontology. Opt. Eng. **54**(5), 053109 (2015). doi:10.1117/1.OE.54.5.053109
6. Arslan, A.N., Anandan, J., Fry, E., Pandey, R., Monschke, K.: A new structure representation for RNA and fast RNA substructure search. In: Proceedings of CSCI 2016, Las Vegas, USA, 14–17 December, pp. 1226–1231. IEEE CPS (2016). doi:10.1109/CSCI.2016.230
7. Aslan, C., Tari, S.: An axis based representation for recognition. In: Proceedings of ICCV, pp. 1339–1346 (2005)
8. Bai, X.Y., Yui, D., Latecki, L.J.: Skeleton-based classification using path similarity. Int. J. Pattern Recogn. Artif. Intell. **22**(4), 733–746 (2008)
9. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. Trans. PAMI **24**, 209–222 (2002)
10. Bronstein, A.M., Bronstein, M.M., Kimmel, R., Mahmoudi, M., Sapiro, G.: A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. Int. J. Comput. Vis. **89**(2), 266–286 (2010)
11. Chekmarev, D.S., Kholodovych, V., Balakin, K.V., Ivanenkov, Y., Ekins, S., Welsh, W.J.: Shape signatures: new descriptors for predicting cardiotoxicity in silico. Chem. Res. Toxicol. **21**, 1304–1314 (2008)
12. Chen, L., Feris, R.S., Turk, M.: Efficient partial shape matching using the Smith-Waterman algorithm. In: Proceedings of CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, Anchorage, Alaska, June 2008. doi:10.1109/CVPRW.2008.4563078
13. Cour, T., Shi, J.: Recognizing objects by piecing together the segmentation puzzle. In: Proceedings of CVPR (2008). doi:10.1109/CVPR.2007.383051
14. Badawy, O., Kamel, M.: Matching concavity trees. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR/SPR 2004. LNCS, vol. 3138, pp. 556–564. Springer, Heidelberg (2004). doi:10.1007/978-3-540-27868-9_60
15. Felzenszwalb, P., Schwartz, J.: Hierarchical matching of deformable shapes. In: Proceedings of CVPR, pp. 1–8. IEEE (2007)
16. Gavrila, D.M.: Pedestrian detection from a moving vehicle. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 37–49. Springer, Heidelberg (2000). doi:10.1007/3-540-45053-X_3
17. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoret. Comput. Sci. **38**(2–3), 293–306 (1985)
18. Grega, M., Matiolański, A., Guzik, P., Leszczuk, M.: Automated detection of firearms and knives in a CCTV image. Sensors **16**(1), 47 (2016). doi:10.3390/s16010047
19. Huttenlocher, D., Klanderman, G., Rucklidge, W.: Comparing images using the Hausdorff distance. Trans. PAMI **15**(9), 850–863 (1993)
20. Huang, R., Pavlovic, V., Metaxas, D.N.: A profile Hidden Markov Model framework for modeling and analysis of shape. In: Proceedings of International Conference on Image Processing, Atlanta, GA. IEEE, October 2006. doi:10.1109/ICIP.2006.312827
21. Kim, H.-S., Chang, H.-W., Liu, H., Lee, J., Lee, D.: BIM: Image matching using biological gene sequence alignment. In: Proceedings of International Conference on Image Processing, pp. 205–108. IEEE ICIP (2009)
22. Kortagere, S., Chekmarev, D., Welsh, W.J., Ekins, S.: Hybrid scoring and shape based classification approaches for human pregnane X receptor. Pharm. Res. **26**(4), 1001–1011 (2009)

23. Kortagere, S., Krasowski, M.D., Sean Ekins, S.: The importance of discerning shape in molecular pharmacology. Trends Pharmacol. Sci. **30**(3), 138–147 (2009)
24. Korotkov, K.: Automatic change detection in multiple skin lesions. Ph.D. thesis, Universitat de Girona (2014)
25. Ling, H., Jacobs, D.: Using the Inner-Distance for classification of articulated shapes. In: Proceedings of Conference on Computer Vision and Pattern Recognition, vol. II, pp. 719–726. IEEE CVPR (2005)
26. Mori, G., Malik, J.: Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In: Proccedings of Computer Vision and Pattern Recognition, pp. 134–141. IEEE CVPR (2003)
27. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**(3), 443–53 (1970)
28. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 575–588. Springer, Heidelberg (2006). doi:10.1007/11744047_44
29. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: Proceedings of ICCV (2005). doi:10.1109/ICCV.2005.63
30. Sirakov, N.M.: A new active convex hull model for image regions. J. Math. Imaging Vis. **26**(3), 309–325 (2006)
31. Sirakov, N.M., Simonelli, I.: A new automatic concavity extraction model. In: Proceedings of SSIAI, pp. 178–182 (2006)
32. Sirakov, N.M., Ushkala, K.: An integral active contour model for convex hull and boundary extraction. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5876, pp. 1031–1040. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10520-3_99
33. Sklansky, J.: Measuring concavity on a rectangular mosaic. IEEE Trans. Comput. **C-21**, 1355–1364 (1972)
34. Sun, K.B., Super, B.J.: Classification of contour shapes using class segment sets. In: Proceedings of CVPR, pp. 727–733 (2005). doi:10.1109/CVPR.2005.98
35. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: Proceedings of CVPR, pp. 127–134 (2003). doi:10.1109/CVPR.2003.1211346
36. Wang, H., Oliensis, J.: Rigid shape matching by segmentation averaging. IEEE Trans. Pattern Anal. Mach. Intell. **32**(4), 619–635 (2010). doi:10.1109/TPAMI.2009.199
37. Xu, X., Lee, D.-J., Antani, S., Long, L.R.: A spine x-ray image retrieval system using partial shape matching. IEEE Trans. Inf Technol. Biomed. **12**(1), 100–108 (2008)