



An interactive motion planning framework that can learn from experience

Yu Yan*, Emilie Poirson, Fouad Bennis

LUNAM Université, IRCCYN, Ecole Centrale de Nantes, France



HIGHLIGHTS

- Maximal sphere sequence represents the scenario. The centerline of maximal sphere sequence represents the scenario skeleton.
- The motion of object has close relationship with the change of the skeleton and volume size of scenario.
- Motion learning is based on computing the similarity of scenarios by dynamic time warping (DTW).
- Scenario retrieval is highly efficient due to the hierarchical clustering in motion library.
- Releasing humans from rotation manipulation in complex assembly verification with learned motion experience.

ARTICLE INFO

Article history:

Received 28 February 2014

Accepted 28 July 2014

Keywords:

Accessibility verification

Assembly path planning

Narrow passage

Interactive motion planning

Scenario retrieval

ABSTRACT

The accessibility verification of the assembly/disassembly plays an important role in the process of product design. In the last decade, the sampling based motion planners have been successfully applied to solve the accessibility verification. However, the narrow passage which is a common problem in the assembly tasks is still a bottleneck. Meanwhile, the requirement of perception and emotion assessment drives the interaction between users and automatic path planners in the virtual assembly process. In this paper, a curve matching method is used to explore the implicit relationship between the topological information of scenarios and the motion of objects, based on which an interactive motion planning framework that can learn from experience is constructed.

Our framework consists of two main processes: a learning process and a motion generation process. In the former process, the motion segment (a part of motion path) and its related scenario segment (a part of workspace passed through by the object) are gathered, after an interactive motion planning process finds a collision-free motion path or reaches the conclusion of inaccessibility. According to the similarity between the skeletons of scenario segments, the gathered scenario segments and motion segments are organized by a hierarchical structure in the motion library. The latter process permits users to control only one point in the workspace for the selection of a new scenario, and then the similar scenarios are retrieved from the motion library, to help quickly detect the connectivity of the new scenario and generate a repaired motion path to guide users with feasible manipulations. We highlight the performance of our framework on a challenging problem in 2D, in which a non-convex object passes through a cluttered environment filled with randomly shaped and located non-convex obstacles.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The verification of accessibility in assembly and disassembly is an essential issue for the product design. However, the accessibility

verification presents challenges to both humans and computers. For humans, the manual manipulation of 3D object in a complex virtual environment is extremely difficult [1]. For computers, the accessibility verification can be defined as computing a collision-free path for an object between a free location and a pre-designed location. In the last decade, the sampling based motion planners have been successfully used to solve the motion planning problems [2]. However, the performance of sampling based planning algorithms may degrade if the collision-free space contains narrow passages [3], i.e. small regions whose removal or perturbation can change the connectivity of the collision-free space [4]. In

* Correspondence to: IRCCyN, 1, rue de la noë, BP 92101, 44321 NANTES CEDEX 3, France. Tel.: +33 6 47 45 56 64.

E-mail addresses: Yu.Yan@ircyn.ec-nantes.fr, yanyu860625@gmail.com (Y. Yan), emilie.boirson@ircyn.ec-nantes.fr (E. Poirson), fouad.bennis@ircyn.ec-nantes.fr (F. Bennis).

particular, the narrow passages commonly exist in the virtual assembly/disassembly tasks. Moreover, if no collision-free path exists, the sampling based algorithms may run for a long time before verifying that [5]. Different heuristics have been created to bias the automatic planners. *Decomposition based planners* decompose the configuration space into different levels of cells and get an approximate corridor with A* algorithm, then explore this corridor to find a collision-free path [6,7]. *Obstacle based planners* either plan motion compliant to the C-contact space (C-contact space is the subset of configuration space, which consists of the configurations when the robot touches one or more obstacles without any penetration [4]) with geometric formulations [8,9], or increase the sampling around the obstacle [10,11] and in the C-contact space [12]. In [13], the vectors generated on the triangulated meshes of CAD models are used to guide the extension of basic RRT planner (Rapidly-exploring Random Trees). *Retraction based planners* attempt to improve the sampling in narrow passages through retracting the in-collision configuration to the nearest configuration in the C-contact space [4,14]. Despite some improvements, the narrow passage is still a major issue for the sampling based planners. The weak consciousness to the local changes of geometric space results in the above difficulties.

Moreover, designers need to test and verify the geometric constraints to the articulate system of a human skeleton. A direct method is adding a digital human mannequin into the reasoning loop of an automatic method [1]. The quality of the generated motion of the mannequin becomes a major issue, which mainly concerns reducing the redundant degrees of freedom and making the motion look more natural. Besides, a key challenge for the designers is to analyze end-users' responses in the design process and to promote product innovation, but users' preferences cannot be precisely formulated [15]. Hence, the optimization of the preference of users is today extremely difficult for an automatic method, and the integration with the direct participation of designers is an essential stage [16,17].

1.1. Interactive motion planners

A solution to the above challenges is to make humans interact with an automatic motion planner. By relaxing the collision constraints, it is easier for a human to find a motion solution. Interactive motion planners attempt to integrate humans with the automatic path planners. The method in [18] enables users with a haptic device to visually specify some critical via-configurations or sub-goals in the workspace, which guide the exploration of a sampling based planner. However, in the practice, some via-configurations in complex environments are unlikely to be found by humans, or unreachable. In [6,19], a preliminary discretization of configuration space is carried out by a decomposition algorithm, and then a corridor between the initial configuration and the goal configuration is found. In this corridor, the harmonic functions are formulated to provide force feedback and a sampling based planner searches a feasible path. Although these kind of methods successfully accelerate the automatic planner by reducing the search space, humans impose very limited impact on the planning process. The method proposed in [20] is successfully applied to the interactive motion planning of a steerable needle, but not applicable to a non-convex object. In [21], the method simply allows the simultaneous cooperation between users and automatic algorithms. But humans and automatic planners cooperate in a relatively independent manner. In the "narrow passages", both of them may fall into a trap and cannot guide each other efficiently.

1.2. Learning from experience in motion planning

Learning experience to serve motion planning in new environments has been studied by various works. Recent research focuses

on exploiting the accumulated data in the prior planning results. In [22], the approach proposes to learn from the prior instances of collision queries and to improve the sampling based motion planners in avoidance of the exact collision checking. The "retrieve and repair" idea of [23] is similar to ours. However, their retrieval process based on the amount of constraints violated by a path is implemented in a robot's configuration space, which limits their adaptability to quite different environments. In [24], a more robust and adaptive motion learning method is proposed to map from the previously optimized trajectories in old situations to the good trajectories in new situations. Although similar to our idea of retrieving paths based on the similarity of workspace, their method is limited to an offline constructed motion library composed of optimized motion trajectories. In contrast, the concept of online motion learning based on the topological similarity between scenarios has been proposed in [25], however, no concrete or workable method was developed by the authors. Given the workspace \mathcal{W} ($\mathcal{W} \subset \mathbb{R}^2$ in 2D; $\mathcal{W} \subset \mathbb{R}^3$ in 3D) and its subset taken up by the obstacles $\mathcal{W}_{\text{obstacle}}$, the scenario for a moving object is the subset of the workspace excluding the obstacles, which is noted as $\mathcal{W}_{\text{scenario}}$, $\mathcal{W}_{\text{scenario}} = \{x|x \in \mathcal{W}, x \notin \mathcal{W}_{\text{obstacle}}\}$. In this paper, the relationship between the topology of scenarios and the motion of objects is discovered, and based on it a novel scenario retrieval and motion reuse strategy is proposed.

1.3. Contributions

In this paper, a novel framework for the interactive motion learning and generation is proposed. The learning is dynamic: the motion library can constantly evolve from an empty one to a very large one, and it is maintained by a balanced hierarchical structure. More detailed contributions include:

- The use of the maximal sphere sequence to represent the scenario. The centerline and the radius change of the sphere sequence can respectively describe the topological structure and the volume change of the scenario. The method of the real time generation of the maximal sphere sequence is also provided.
- A novel motion learning framework based on computing the similarity of the topological structure and the volume change of scenarios.
- The great acceleration of the motion planning process, especially in narrow passages, thanks to the adequate learning and motion generation strategy.
- Simplified ways of interacting with a computer by releasing humans from the rotational manipulations, which make it possible to use non-haptic devices in a complex assembly/disassembly task.

We have applied our algorithms and demonstrated its effect in a challenging case in 2D. The rest of the paper is organized in the following manner. Section 2 briefly surveys the related tools in other domains. Section 3 is dedicated to the detailed explanation of our framework. Simulation results are presented and analyzed in the Section 4. The limitations and extensions are discussed in the Section 5.

2. Related tools from other domains

The recent developments in the fields of motion primitive, medial axis, curve skeleton and curve matching are the basis of our motion learning and generation framework.

2.1. Motion primitive

The concept of motion primitive is similar to our idea of saving motion segments into motion library. Motion primitives have

been widely used to plan the complex motions of virtual mannequins and humanoid robots. These motion primitives can be represented as the optimized stances [26,27], or the differential equations called Dynamic Movement Primitives (DMPs) [28]. The adoption of motion primitives aims to reduce the redundant degrees of freedom and decrease the search space. The motion primitive established on experience is an imitation of motion learning mechanism of humans. However, to our knowledge, this work has never been used to plan a general object's motion. One of the challenges is to adaptively decide which primitive should be used [26]. Here, we provide a new solution to this challenge by computing the topological similarity between scenarios. The medial axis and curve skeleton are robust tools describing the topological structure of shapes in 2D and 3D respectively.

2.2. Medial axis and curve skeleton

The medial axis is a lower dimensional representation of objects, which is defined by the locus of the maximal balls tangent to the object surface at two or more points [29]. Due to its topological representation of a shape, it has been widely used for biological shape description, computer vision and solid modeling. In the motion planning field, it is mainly used for translational exploration in 2D plane [30]. In 3D application, the sampled configurations are retracted onto the medial axis of the collision-free configuration space [31,32]. However, this method also endures some expensive computations like other retraction based motion planners.

Curve skeletons are the curves in 1D, which are locally centered in a shape, and capture the essential topology of the underlying object in an understandable and very compact form [33]. The applications of the curve skeleton include: virtual navigation, registration, animation, morphing, scientific analysis, shape recognition, and shape retrieval [34]. There are many different curve skeleton extraction algorithms, and for their brief overview one can refer to [35]. In the shape recognition field, the curve skeleton is usually applied to describe the articulate systems of objects [36]. Since the topology of a scenario in this paper is the medial axis and the curve skeleton with branches pruned, the scenario recognition is actually a curve matching problem.

2.3. Curve matching

Curve matching is one of the central issues in computer vision, which has broad applications in time series analysis [37], speech recognition [38], signature verification [39], and similarity search in the trajectory database [40]. Curve matching attempts to find a distance metric measuring the similarity between two selected curves. Usually, the two compared curves are represented as discrete sequences. A category of algorithms uses dynamic programming to find the correspondence by the shortest path on an accumulated distance matrix, and measures the similarity by the length of this path. The representatives of these algorithms include Dynamic Time Warping (DTW) [41], Longest Common Sub Sequence (LCSS) [42] and Edit Distance on Real sequences (EDR) [43]. An exact indexing method of DTW has been demonstrated in [44], in which a piecewise aggregate lower bounding technique was used to model a sequence with a linear combination of box basis functions. Due to the intensive studies on it, DTW is adopted in this paper.

3. Interactive motion learning and generation

In this section, a hypothesis is proposed to describe the relationship between the motion of an object and its corresponding scenario. Then an overview of our motion learning and generation

framework is demonstrated prior to the detailed explanations of each functional module.

3.1. Relationship between the motion of an object and its scenario's topology

The configuration of a moving object with n DOFs can be represented as a point in an n -dimensional *configuration space* $\mathcal{C}\mathcal{S}$. A configuration q is free if the moving object placed at q does not collide with the obstacles or with itself. The collision-free space $\mathcal{C}\mathcal{S}_{\text{free}}$ is defined as the set of all free configurations in $\mathcal{C}\mathcal{S}$. The computation of the complete $\mathcal{C}\mathcal{S}_{\text{free}}$ by the adaptive sampling has been studied in [45,46]. However, computing the complete $\mathcal{C}\mathcal{S}_{\text{free}}$ cannot fulfill the efficiency requirement. Hence the sampling based motion planners search out a collision-free motion path on a graph which is established by randomly taking samples in $\mathcal{C}\mathcal{S}$ and connecting the adjacent samples with collision-free edges. The performance of the sampling strategy degrades in the narrow passages of $\mathcal{C}\mathcal{S}_{\text{free}}$. An ideal solution to this problem is to find a low-cost function f which can approximately compute $\mathcal{C}\mathcal{S}_{\text{free}}$ from the workspace $\mathcal{W}\mathcal{S}, f : \mathcal{W}\mathcal{S} \rightarrow \mathcal{C}\mathcal{S}_{\text{free}}$. Unfortunately, the computational complexity of f is not lower than that of $\mathcal{C}\mathcal{S}_{\text{free}}$. In another perspective, for a given object, a mapping f_1 from the current scenario $\mathcal{W}\mathcal{S}_{\text{scenario}}^1$ to the Boolean value of accessibility can be found:

$$f_1 : \mathcal{W}\mathcal{S}_{\text{scenario}}^1 \rightarrow b, \quad b \in \{\text{accessible, inaccessible}\}. \quad (1)$$

If b is “accessible”, there also exists a mapping f_2 from $\mathcal{W}\mathcal{S}_{\text{scenario}}^1$ to a collision-free path p :

$$f_2 : \mathcal{W}\mathcal{S}_{\text{scenario}}^1 \rightarrow p, \quad p \subset \mathcal{C}\mathcal{S}_{\text{free}}. \quad (2)$$

Thus the problem is that, for a new scenario $\mathcal{W}\mathcal{S}_{\text{scenario}}^2$, if there exists a dissimilarity distance metric:

$$D : \mathcal{W}\mathcal{S}_{\text{scenario}} \times \mathcal{W}\mathcal{S}_{\text{scenario}} \rightarrow d, \quad d \in R \quad (3)$$

and $D(\mathcal{W}\mathcal{S}_{\text{scenario}}^1, \mathcal{W}\mathcal{S}_{\text{scenario}}^2) < \varepsilon$ where ε is a threshold, is it possible to apply the path p to the new scenario $\mathcal{W}\mathcal{S}_{\text{scenario}}^2$? Or if the mapping f_2 fits a set of scenarios which are similar to $\mathcal{W}\mathcal{S}_{\text{scenario}}^1$. In Fig. 1, a motion path of an “L” shape object is applied to three tightly constrained scenarios, even though the shapes of these scenarios seem quite different. This empirical evidence leads to our hypothesis that the motion path p can be applied to a set of scenarios which are similar to each other, although the concept of “similar” has not yet been defined. In Fig. 1, the narrowest part of the three scenarios is cut out and demonstrated on their right side. The medial axis constructed according to the boundary of obstacles describes the topological structure of the three scenario segments. Intuitively, the medial axes of the three scenario segments are similar to each other when pruning the branches. The medial axis with branches pruned equals the centerline of the maximal sphere sequence. The C-contact space $\mathcal{C}\mathcal{S}_{\text{contact}}$ of the three scenarios is computed through uniformly sampling the configuration space, and is shown by the blue surface in Fig. 1. It is obvious that the topological structure of $\mathcal{C}\mathcal{S}_{\text{free}}$ is dramatically influenced by the topological structure of the scenario. Thus, the similarity between scenarios can be measured by comparing their topological structures. It is also noticed that the radius change of the sphere sequence in the three scenarios shows the similar trend. The fact that the object has much less options of motion in a small volume than in a large volume is the source of the sampling based planners' trouble in narrow passages. On the contrary, the more tightly the scenario is constrained, the stronger the correlation between the motion of an object and the topological structure of its corresponding scenario is, and that turns into the foundation of our motion leaning framework.

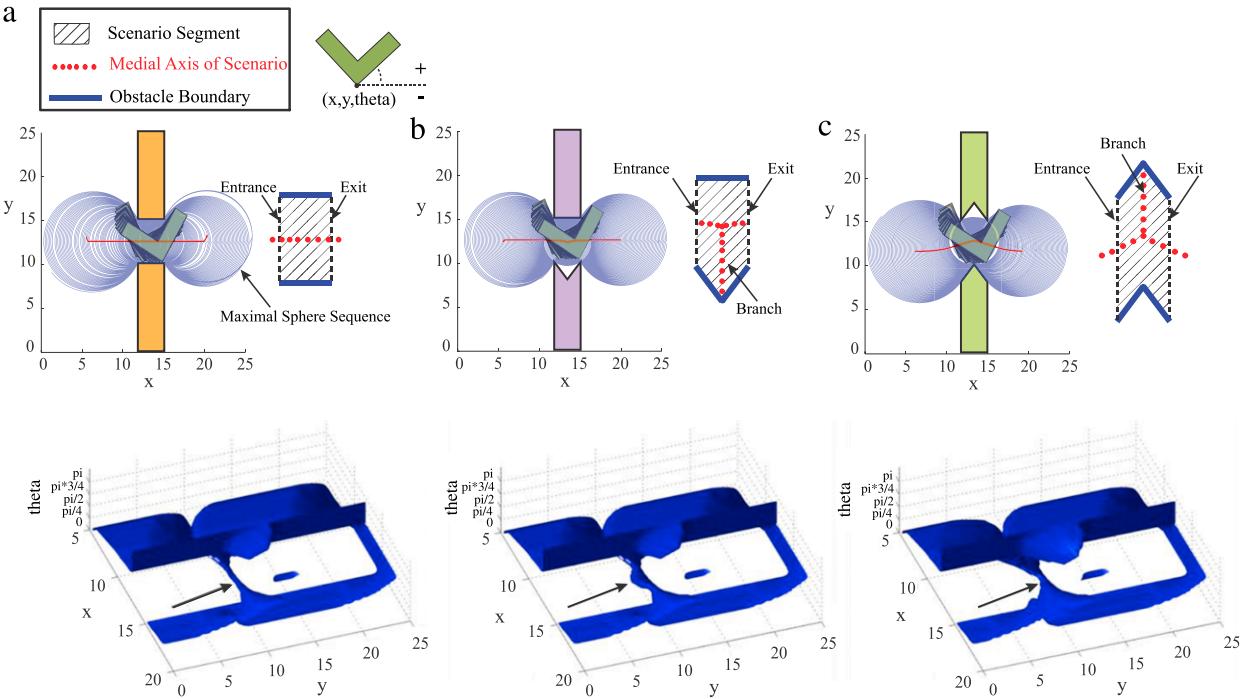


Fig. 1. Similar scenario segments and their topological representations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

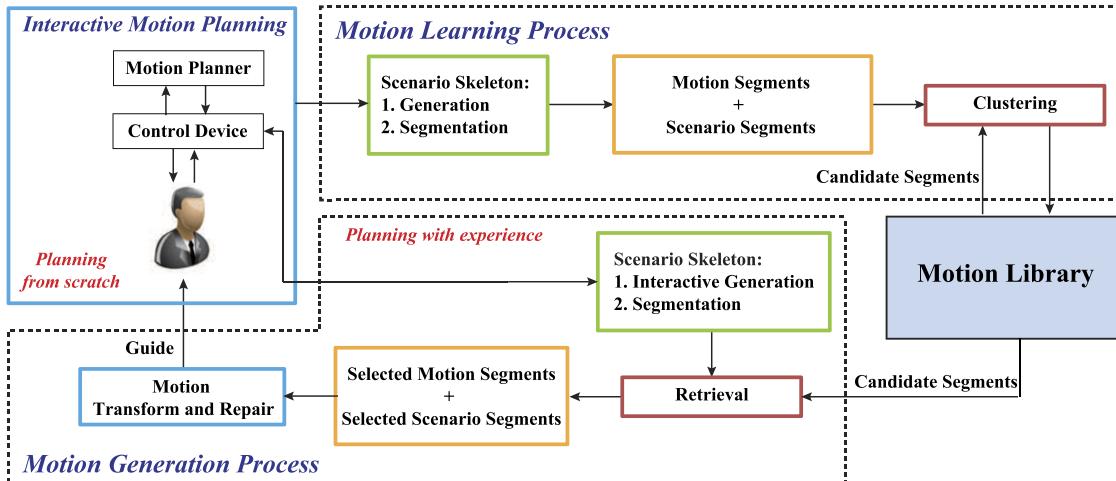


Fig. 2. Framework overview.

3.2. Framework overview

Fig. 2 shows a global view of our interactive motion learning and generation framework. Our system consists of four modules:

- (1) *Interactive motion planning*: this is a planning-from-scratch module which runs when there is no experience or not enough experience. The interactive motion planning method in [25] is used as this module. In fact, our framework is compatible with any kind of interactive motion planning method.
- (2) *Motion learning process*: every time when the accessibility or inaccessibility is verified after an interactive motion planning process, a new process is started to learn from the motion experience by clustering the scenario segments and their motion segments into a motion library.
- (3) *Motion generation process*: with this module, users can firstly select a scenario by controlling one point in workspace. Then

the subsequent functions generate new motions in this scenario with the learned experience.

- (4) *Motion library*: scenario segments with their motion segments as experience are recorded and organized by a hierarchical structure in the motion library.

3.3. Motion learning process

Usually a collision-free motion path or a non-existence conclusion is obtained after an interactive motion planning process. The subsets of workspace passed through by the object constitute the motion scenario. Then the scenario skeleton is generated to represent the topological structure of the scenario. Considering the efficiency and robustness of the scenario recognition, the scenario and its skeleton are split into segments. And every scenario segment is related to a motion segment. According to the similarity, the newly

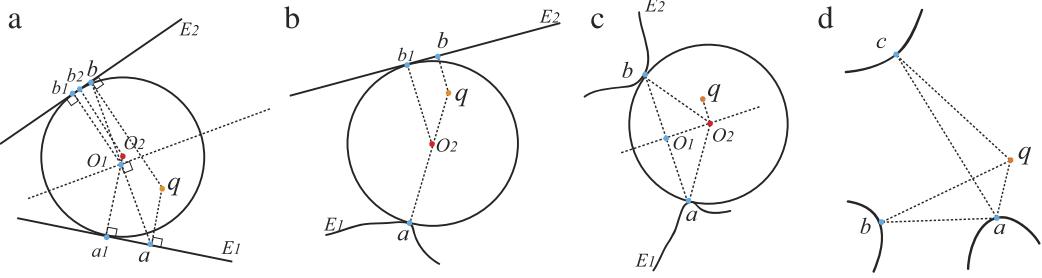


Fig. 3. Sphere generation according to the position of the reference point.

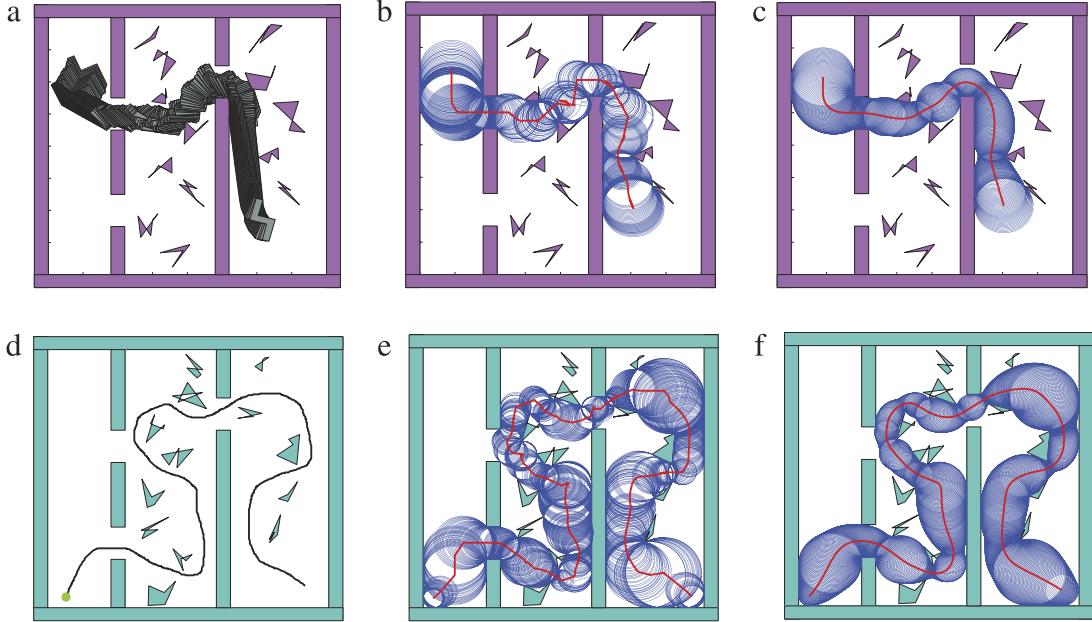


Fig. 4. Sphere generation in a planar cluttered environment in two cases.

gathered scenario segments including their motion segments are clustered into a motion library.

3.3.1. Scenario skeleton generation

The scenario skeleton is defined as the centerline of the maximal local spheres that are tangent to the surrounding obstacles with at least two points. The scenario skeleton can be also regarded as the medial axis in 2D or the curve skeleton in 3D with branches pruned. Calculating the medial axis and curve skeleton for a whole workspace is time-consuming, which is far beyond the requirement of real-time interactions. In this paper, only the scenarios passed through by the target object or selected by users are concerned.

In our method, every local sphere is constructed based on the relative distances between a reference point and the nearest surfaces, edges or vertices of obstacles. The reference point could be the point on the object in the motion learning process or the point manipulated by users in the motion generation process. In Fig. 3, the generation of spheres in three situations is demonstrated with pairs of the nearest surface-surface, surface-vertex and vertex-vertex. The edge and surface serve the equal function.

(1) *Surface-surface*: in Fig. 3(a), the collision detection algorithm finds out two nearest surfaces $\{E_1, E_2\}$ to the reference point q . The points $\{a, b\}$ on the surfaces E_1 and E_2 are the perpendicular foots from the point q . From the center point O_1 between a and b , two perpendicular foots $\{a_1, b_1\}$ on the surfaces E_1 and

E_2 can be also found. Through simple vector operations, a local sphere with center O_2 which is tangent to the surfaces E_1 and E_2 on a_1 and b_2 is created.

- (2) *Surface-vertex*: in Fig. 3(b), the surface E_2 and the vertex a on the surface E_1 are the nearest to the reference point q . The point b on the surface E_2 is the perpendicular foot from the point q . Through simple vector operations, a local sphere with center O_1 which is tangent to the surfaces E_1 and E_2 on a and b_1 is created.
- (3) *Vertex-vertex*: in Fig. 3(c), two vertices ab on the surface E_1 and E_2 are the nearest to the reference point q . On the line which is vertical to the line ab and passes the center point O_1 of the line ab , the center of sphere O_2 which is the perpendicular foot from the point q is found. The sphere is tangent to the surfaces E_1 and E_2 on a and b .

In the situation shown in Fig. 3(d), the perpendicular foot on the line ab from q is not in the middle of a and b . The maximal tangent sphere cannot be computed by the three methods above, thus the third nearest point c is selected to replace b .

Fig. 4 shows the generation of the scenario skeleton in two cases. The first case happens after a motion path is found in Fig. 4(a), in which the spheres are generated according to the trace of a vertex of the object in Fig. 4(b). In the second case, according to the trace of the mouse's cursor which is manipulated freely by a user shown in Fig. 4(d), a sphere sequence is generated in Fig. 4(e). All the centers of spheres constitute the skeleton of a scenario. The

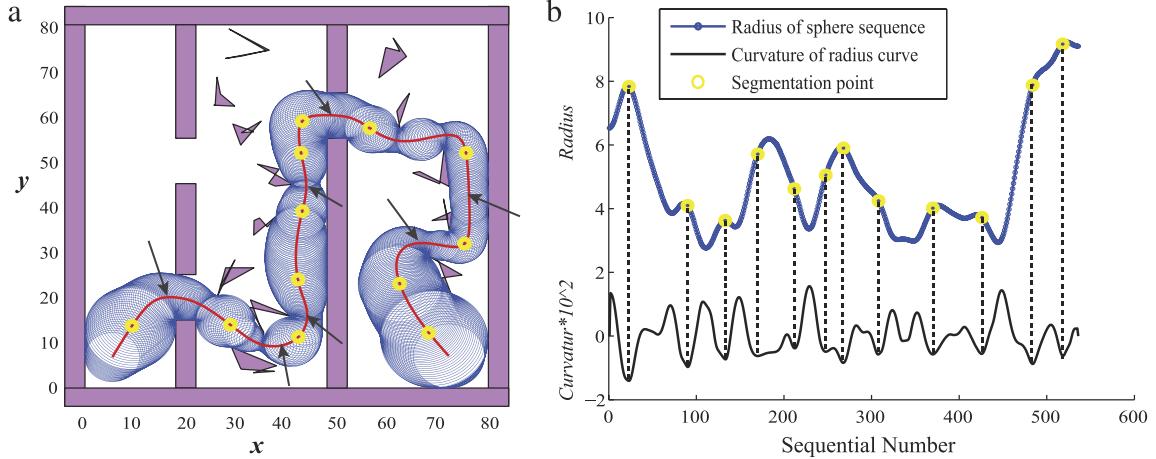


Fig. 5. The segmentation of a scenario.

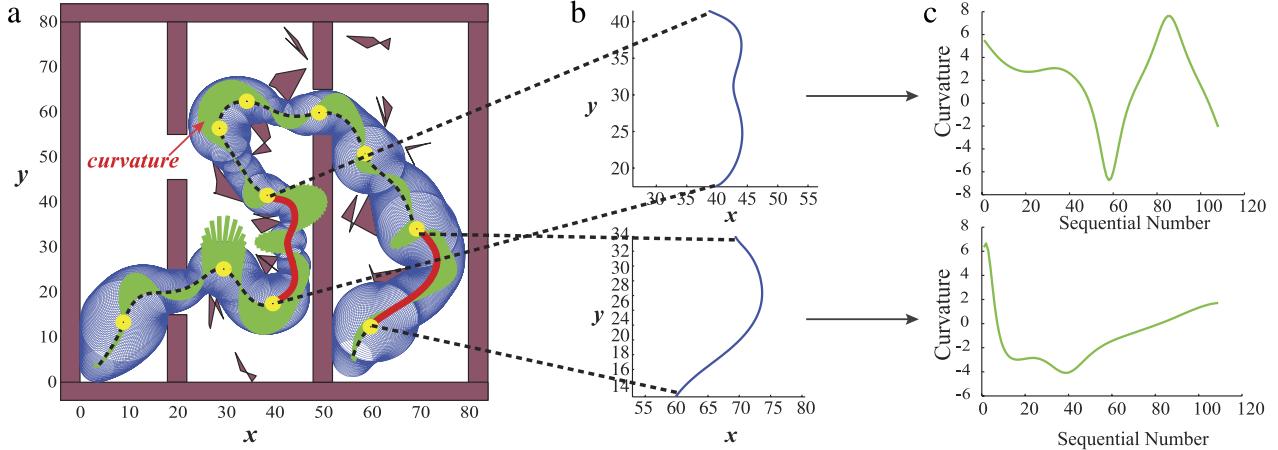


Fig. 6. Calculation of the scenario skeleton's curvature.

jitters on the skeleton may damage the scenario matching which will be introduced in the following sections, thus the smoothing is essential, shown in Figs. 4(c) (f). Actually, the generated scenario skeleton is an approximate medial axis or curve skeleton. Although our method trades off accuracy for speed in order to satisfy the interaction frequency, the accuracy is enough for our application.

3.3.2. Scenario segmentation

The generated scenario skeletons are of different lengths. The partial matching between a local scenario and a global scenario is time-consuming at present. In our method, the scenario is split into segments for the robustness and parallel processing. The segmentation is based on the changing trend of the spheres' radius. The narrow passages usually appear when the radius of the maximal sphere shrinks firstly and then re-enlarge. The beginning and ending locations of this kind of wave can be detected by the curvature of the radius curve. For each point of a planar curve, the curvature is the reciprocal of its osculating circle's radius.

For a plane curve given parametrically in Cartesian coordinates as $\gamma(t) = (u(t), v(t))$, the curvature k is

$$k = \frac{u'v'' - v'u''}{(u'^2 + v'^2)^{3/2}}. \quad (4)$$

The method in [47] is adopted to calculate the curvature, which firstly fits a quadratic polynomial to the local points and then finds out the derivatives of u and v . In the rest of this paper, the same curvature calculation is adopted. In Fig. 5(a), the segmentation of a

scenario is demonstrated. The radius change of the spheres is shown in Fig. 5(b) and its curvature change is shown below it. In the figures, the segmentations always take place at the bottom of a valley on the curvature curve. Given the threshold ε set by users, if the bottom of a valley is lower than ε , the corresponding point on the radius curve is taken as a segmentation point. Then the segmentation locations are projected to the scenario and its skeleton. The scenarios which may produce the narrow passages are cut out by the segmentation and indicated by the black arrows in Fig. 5(a).

3.3.3. Scenario matching

The scenario segments in motion library are quite different in location and orientation. The similarity between scenarios is measured by the distance in the feature space \mathcal{F} . To guarantee the adaptability of the similarity measurement, the features should be invariant to the Euclidean transformation, namely the rotation and translation. Under the scaling and shearing, the relation between the motion of an object and its scenarios is still not clear. The curvature of the scenario skeleton and the radius change of the scenario are the two features which are rotational and translational invariant. In Fig. 6(b), the skeletons of the two scenario segments selected from Fig. 6(a) are zoomed, and their curvature curves are shown in Fig. 6(c).

The curvature and radius curves are sampled into two sequences. An appropriate distance metric is required to measure the similarity between the curvature sequences and the similarity between the radius sequences. Besides, the distance metric is

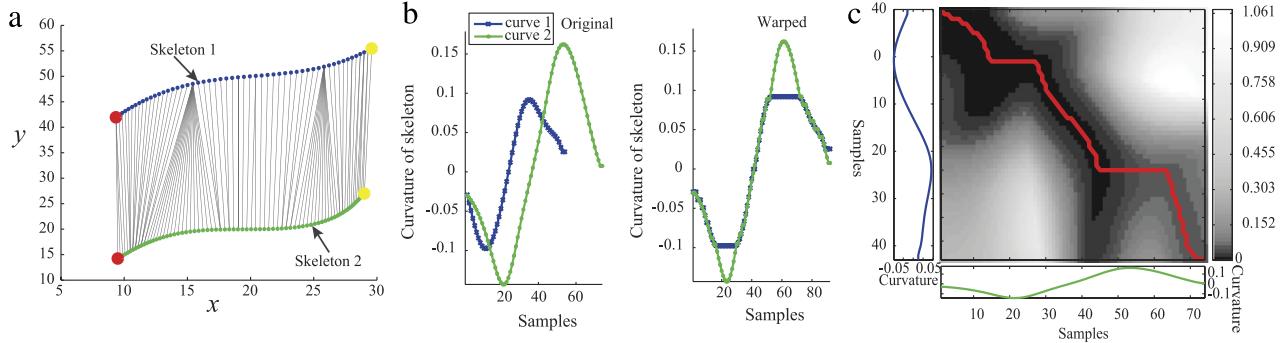


Fig. 7. The alignment between two similar scenario skeletons by DTW. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

also required to provide the accurate alignment between two sequences, which serves the motion transformation phase. Dynamic time warping (DTW) is a well-known technique to find the optimal alignment between two given sequences under certain restrictions. Intuitively, the sequences are warped in a nonlinear fashion to match each other. Indeed, any data which can be turned into a linear sequence can be analyzed with DTW [38].

The objective of DTW is to compare two sequences: $X := (x_1, x_2, \dots, x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, \dots, y_M)$ of length $M \in \mathbb{N}$, $x_n, y_m \in \mathcal{F}$ for $n \in [1 : N]$ and $m \in [1 : M]$. In our application, there are two kinds of sequences, one is the curvature change of scenario skeleton and the other is the radius change of sphere sequence. Thus, the value in \mathcal{F} can be the curvature or radius. For the two scenario skeletons in Fig. 7(a), whose starting and ending points are marked with yellow and red colors, the change of their curvature before and after the alignment by DTW is shown in Fig. 7(b). For every element x_n in X , we can find its Euclidean distance to every element y_m in Y , which constructs a local cost matrix $C \in \mathbb{R}^{N \times M}$ defined by $C(n, m) := c(x_n, y_m)$. The distance between the two skeletons is measured by the total cost of an optimal path from the initial point $(1, 1)$ to the final point (N, M) in the cost matrix C , which is a nonlinear warping between the two skeletons. The dynamic time warping is calculated by the same definition and formulation in [38].

An (N, M) warping path is a sequence $p := (p_1, p_2, \dots, p_L)$ with $p_l = (n_l, m_l) \in [1 : N] \times [1, M]$ and $l \in [1 : L]$ satisfying the three conditions:

- (1) Boundary condition: $p_1 = (1, 1)$ and $p_L = (N, M)$.
- (2) Monotonic condition: $n_1 \leq n_2 \leq \dots, n_L$ and $m_1 \leq m_2 \leq \dots, m_L$.
- (3) Step size condition: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in [1 : L - 1]$.

An (N, M) warping path $p := (p_1, p_2, \dots, p_L)$ defines an alignment between two sequences $X := (x_1, x_2, \dots, x_N)$ and $Y := (y_1, y_2, \dots, y_M)$ by assigning the element x_{n_l} of X to the element y_{m_l} of Y . The total cost $c_p(X, Y)$ of a warping path p between X and Y with respect to the local cost measure c is defined as:

$$c_p(X, Y) = \sum_{l=1}^L c(x_{n_l}, y_{m_l}). \quad (5)$$

An optimal warping path between X and Y is the warping path p^* having minimal total cost among all possible warping paths. The DTW distance $DTW(X, Y)$ between X and Y is defined as the total cost of p^* :

$$DTW(X, Y) = c_{p^*}(X, Y) \\ = \min\{c_p(X, Y) | p \text{ is an } (N, M) \text{ warping path}\}. \quad (6)$$

The optimal warping path can be found by the dynamic programming technique, which has to construct an accumulated

matrix at first. The accumulated matrix D is constructed as follows:

- (1) $D(n, 1) = \sum_{k=1}^n c(x_k, y_1)$ for $n \in [1 : N]$,
- (2) $D(1, m) = \sum_{k=1}^m c(x_1, y_k)$ for $m \in [1 : M]$,
- (3) $D(n, m) = \min\{D(n - 1, m - 1), D(n - 1, m), D(n, m - 1)\} + c(x_n, y_m)$, $1 < n \leq N$ and $1 < m \leq M$.

Actually, the $DTW(X, Y) = D(N, M)$ can be computed with $O(N, M)$ operations. The accumulated matrix computed from the local cost matrix is shown in Fig. 7(c). The optimal warping path is computed in the reverse order of the indices starting with $p_L = (N, M)$. The pseudo code is shown in Algorithm 1. The optimal warping path computed by this algorithm is shown by the red curve in Fig. 7(c), which indicates the optimal alignment of the two skeleton segments in Fig. 7(a). It can be noticed that there are two horizontal lines on the optimal warping path in Fig. 7(c), in each of which a single element of the first skeleton is aligned to many consecutive elements of the second skeleton shown in Fig. 7(a).

In Fig. 8, for two specific query scenario segments, six scenario segments having the most similar skeletons are retrieved from the motion library (100 segments) by DTW. The result is sorted according to their dissimilarity distances which are labeled on top of them with their ID numbers in the motion library. For example, the scenario segments No. 85 and No. 62 have the most similar skeletons to the query scenario segment. The result also reveals that two scenario segments with similar bending skeletons may have quite different radius changes, which may cause completely different motions. Thus, in order to find a completely similar scenario to a query scenario, both the curvature and radius need to be considered.

Algorithm 1: Optimal Warping Path

```

1 Input: Accumulated cost matrix  $D$ ;
2 Output: Optimal warping path
    $p^* = (p_1, \dots, p_l, \dots, p_L)$ ;
3    $l \leftarrow L$ ,  $p_l = (n, m) \leftarrow (N, M)$ 
4 while  $l > 1$  and  $p_l \neq (1, 1)$  do
5   if  $n = 1$  then
6      $p_{l-1} \leftarrow (1, m - 1)$ 
7      $l \leftarrow l - 1$ 
8   else if  $m = 1$  then
9      $p_{l-1} \leftarrow (n - 1, 1)$ 
10     $l \leftarrow l - 1$ 
11   else
12      $p_{l-1} = \operatorname{argmin}\{D(n - 1, m - 1), D(n - 1, m),$ 
         $D(n, m - 1)\}$ 
13   end if
14 end while

```

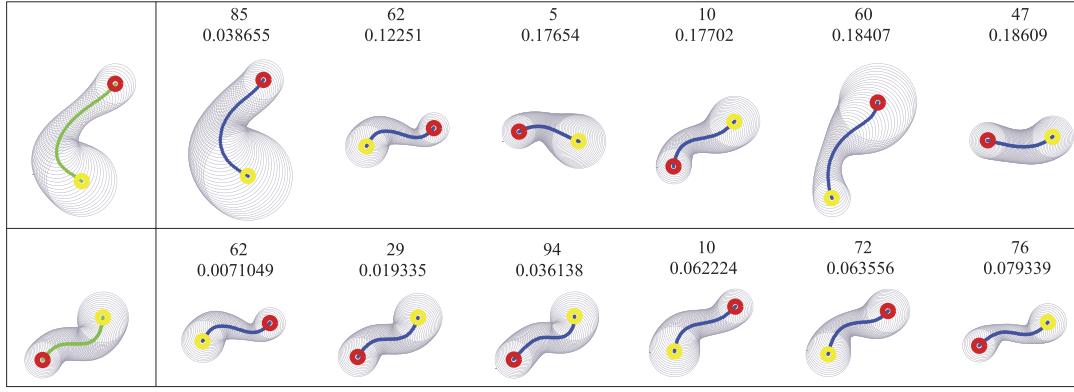


Fig. 8. Retrieval results by DTW using skeleton's curvature feature.

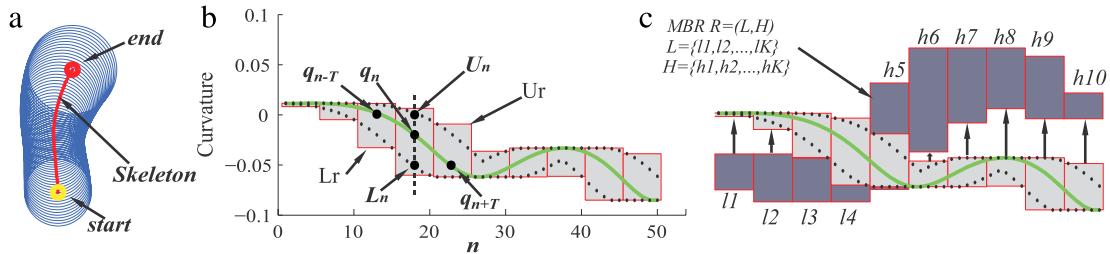


Fig. 9. Piecewise aggregate approximation (PAA) of the bounding constraints and the distance calculation between a skeleton and a minimum bounding rectangle (MBR).

3.3.4. Motion clustering

At the final phase of the learning process, scenario segments and their motion segments are gathered into the motion library. It is important to note that the inaccessible scenario segments are also gathered, because they provide the information to quickly determine the inaccessibility of a new scenario. The motion library gradually grows with the accumulated experience. When it is very large, the sequential indexing method is time-consuming. The indexing of sequences has been widely studied since [37], in which the method uses Discrete Fourier Transformation (DFT) to map a time serial to a point in the multidimensional space and clusters these points with the R-tree algorithm. R-trees are the tree data structures used for indexing multidimensional information. In an R-tree, each object is bounded by a minimum bounding rectangle (MBR), and the higher level node is actually an MBR with larger size which contains many lower nodes (MBR) with smaller size. The R-tree is also a balanced search tree and supports the dynamical insertion and deletion. The searching algorithms are rather simple which use the overlap between MBRs to decide whether or not to search into a subtree. To accelerate indexing, the R-tree is used to organize the scenario segments.

Unfortunately, however, DTW does not obey the triangular inequality and thus has resisted attempts at exact indexing. The exact multidimensional indexing of DTW is studied in [44]. The authors proposed using the piecewise aggregate approximation (PAA) of the global bounding constraints to transform a sequential curve into a multidimensional representation, which makes it possible to use the multidimensional indexing structure for DTW. This method is used here for constructing the indexing structure of scenario skeleton. For a scenario segment in Fig. 9(a), a bounding envelope with the *Upper U* and *Lower L* constraints (dashed lines) is used to enclose the curvature curve of the skeleton in Fig. 9(b).

The envelope limits the optimal warping path found by DTW in a band range along the query curve. Given the curvature curve $Q := (q_1, \dots, q_l, \dots, q_N)$ of a scenario skeleton, the *Upper U* and *Lower L* constraints are defined as follows:

$$\begin{aligned} U_n &= \max(q_{n-T} : q_{n+T}) \\ L_n &= \min(q_{n-T} : q_{n+T}), \quad n \in [1 : N] \end{aligned} \quad (7)$$

where T is a constant value (set as 5 in our application). Then the PAAs of U and L are created, which are U_r and L_r shown in Fig. 9(b). U_r and L_r are defined as follows:

$$\begin{aligned} U_{ri} &= \max\left(U_{\frac{N}{K}(i-1)+1} : U_{\frac{N}{K}i}\right) \\ L_{ri} &= \min\left(L_{\frac{N}{K}(i-1)+1} : L_{\frac{N}{K}i}\right), \quad i \in [1 : K], \quad 1 \leq K \leq N. \end{aligned} \quad (8)$$

The PAA can reduce a curvature curve of length N into K dimensions, where $1 \leq K \leq N$. In our application, the N and K are set as 50 and 10 respectively. When indexing, the distance $Dist$ between the query curvature curve Q and a node of the R-tree is the non-overlapping parts between Q 's (U_r, L_r) and the node's MBR (R), which is the square rooted sum of the squared length of arrow lines in Fig. 9(c). Let $R = (L, H)$, where $L = (l_1, l_2, \dots, l_K)$ and $H = (h_1, h_2, \dots, h_K)$, the distance $Dist$ is defined as follows:

$$Dist(Q, R) = \sqrt{\sum_{i=1}^K (d_i)^2}, \quad d_i = \begin{cases} l_i - U_{ri}, & \text{if } l_i > U_{ri} \\ h_i - L_{ri}, & \text{if } h_i < L_{ri} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Descending from the root node, the query scenario segment is compared iteratively with MBR until a leaf node which is the best size MBR enclosing the query scenario segment is found. Then the query scenario segment is saved into this MBR. If the number of scenario segments in this MBR overflows, this MBR will be split and the structure of motion library will be modified according to the R-tree rules. The clustering process is shown in Fig. 10.

Most of the scenario segments will not be visited in a single search process due to the hierarchical structure of the motion library. Moreover, the search until the leaf nodes only calculates the distance between (U_r, L_r) and MBR by the Euclidean metric. Because the computation of the Euclidean metric is much faster than DTW, the search in the motion library can be very fast. This will be proved by the experimental results. In the motion library, the Successful Scenarios (scenario with collision-free motion path) and the Failure Scenarios (scenario without collision-free motion path) are respectively clustered into two independent trees, named as the Successful Tree and the Failure Tree.

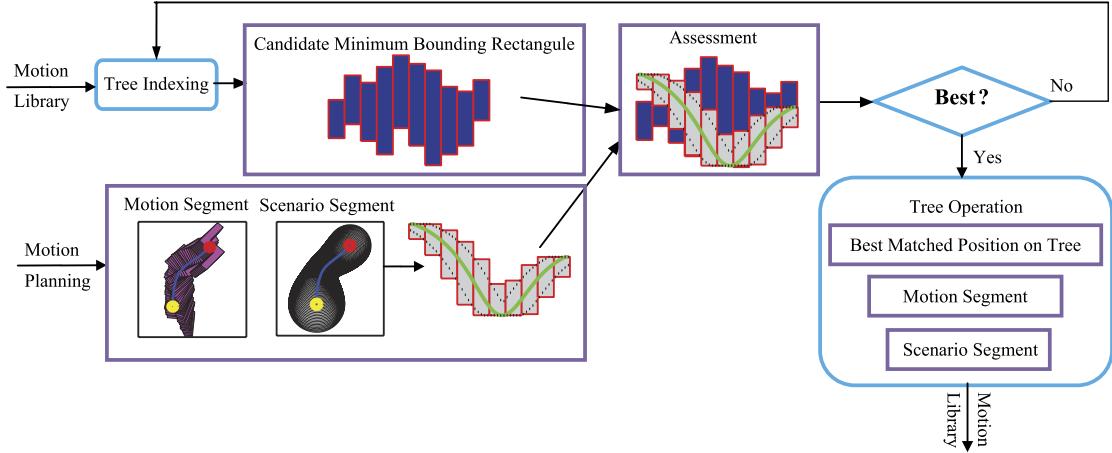


Fig. 10. The process of clustering scenario segments into a motion library.

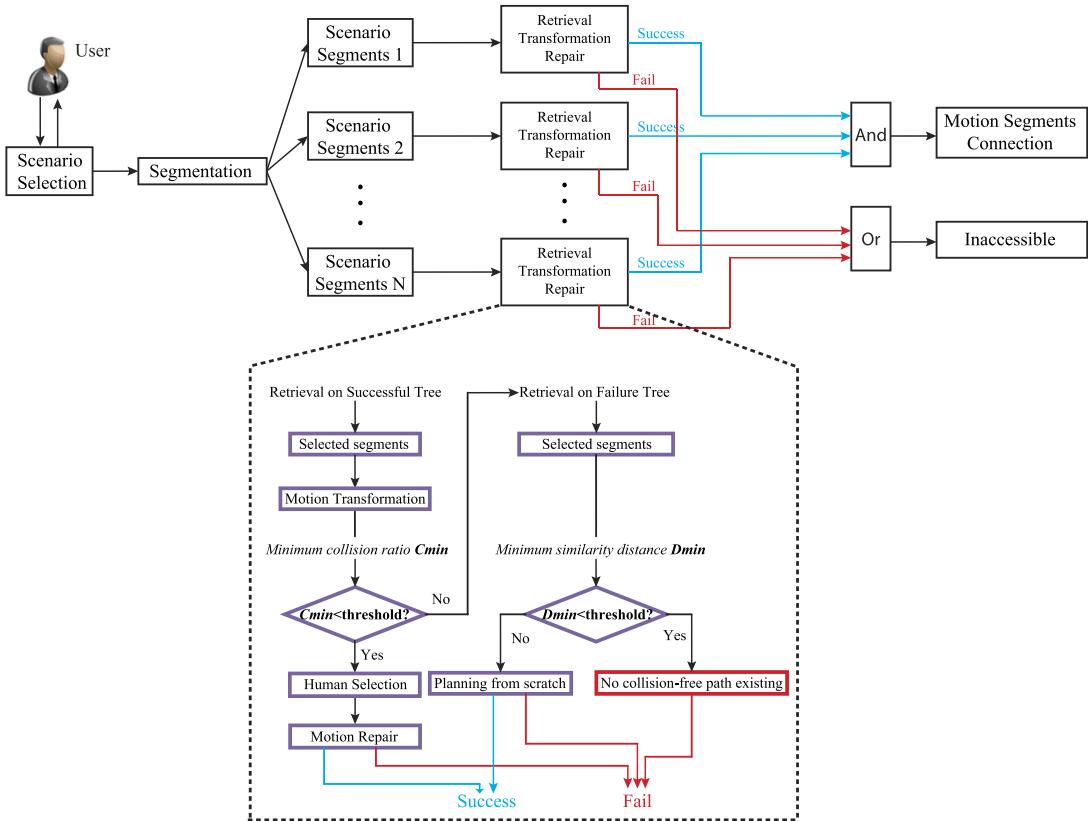


Fig. 11. Motion generation process.

3.4. Motion generation

The data flow of the motion generating process is described in Fig. 11. At the initial phase, users can interactively select their interested scenarios by manipulating a reference point moving in workspace. According to the location of the reference point, the local maximal sphere is generated by the method of Section 3.3.1. The interesting scenario is what the user wants the object to pass through. Since the interaction process only involves translational manipulation, it can greatly reduce the difficulties emerging in the prior interactive motion planning methods, and can more effectively take advantage of user's global planning ability. Then, the selected scenario is split into multiple segments. For each new scenario segment, the most similar scenario segments are retrieved from the motion library. The retrieval process will be firstly

implemented on the Successful Tree. The retrieved motion segment is transformed to fit the new scenario segment according to the alignment of the scenario skeletons. If the transformed motions satisfy the condition with a reasonable quantity of collision, the process continues to repair the imperfect motion segment; otherwise, the retrieval process will be implemented on the Failure Tree. If a new scenario segment is similar to a scenario segment in the Failure Tree, it comes to an “inaccessible” conclusion of it. Otherwise, the accessibility of the new scenario segment is not possible to be detected by experience and the motion generation degrades to the planning-from-scratch. The algorithms used for the planning-from-scratch module can be any kind of interactive or automatic motion planners in the literature. If all of the new scenario segments are accessible, the following process connects all pairs of adjacent motion segments by an automatic motion planner. The

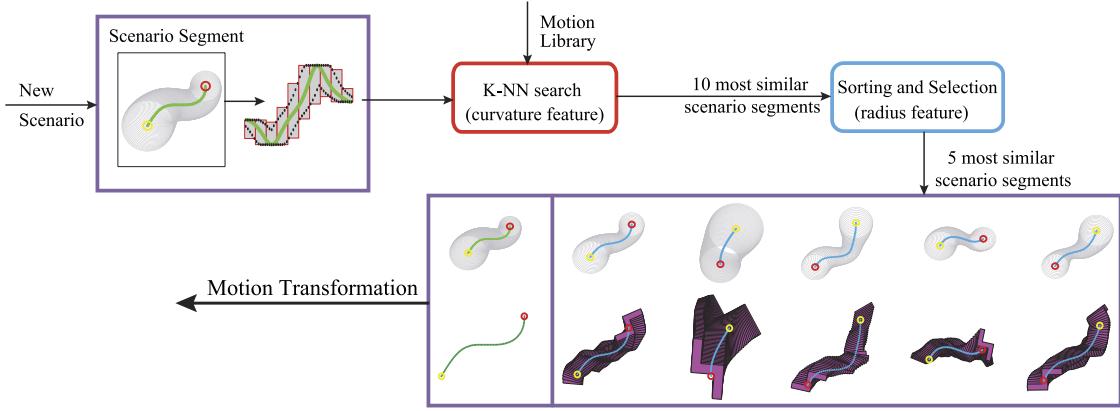


Fig. 12. Scenario retrieval process.

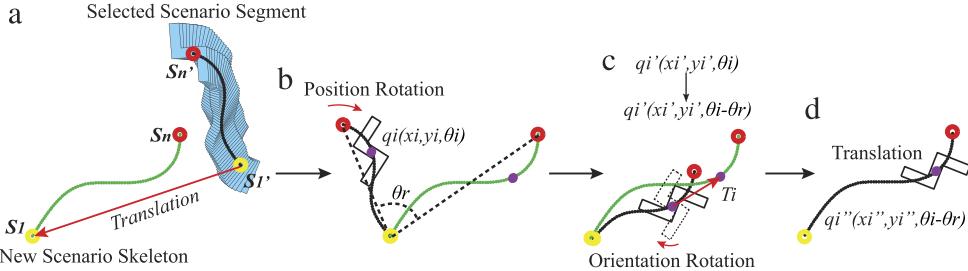


Fig. 13. Motion transformation according to the skeleton transformation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

connection between motion segments runs at great speed, which is always negligible.

3.4.1. Motion retrieval

Retrieval on the Successful Tree: for each scenario segment, the retrieval process firstly makes a K-NN (K Nearest Neighbor) search in the motion library using the curvature of scenario skeleton. The search outside MBR calculates the distance between MBR and (U_r, L_r) . The search inside MBR uses DTW to calculate the distance. After a K-NN search, the ten scenario segments with the most similar bending skeleton to the new scenario segment are retrieved. Then, their similarities to the new scenario segment are re-measured by DTW using the radius feature. Finally, the five highest ranking scenario segments are output to the motion transformation module, see Fig. 12.

Retrieval on the Failure Tree: if none of the five transformed motion segments fulfill the requirement of generating a valid motion, there might be two situations: one is that the motion library is not large enough, in which case the new motion is planned from scratch; the other is that the new scenario segment is not feasible for the object to pass through. Hence the retrieval is continued on the Failure Tree to exclude the latter situation. The search on the Failure Tree is the same as that on the Successful Tree. If the new scenario segment is similar in a predefined range to one scenario segment in the Failure Tree, it can be concluded that this new scenario is impossible to pass through.

3.4.2. Motion transformation and reparation

Motion transformation: the motion segment is transformed according to the alignment between two scenarios' skeletons. Given the skeleton of a new scenario $Sk_{New} = \{s_1, s_2, \dots, s_n\}$, the skeleton and motion of the selected scenario are $Sk_{sel} = \{s'_1, s'_2, \dots, s'_n\}$ and $M_{sel} = \{q_1, q_2, \dots, q_n\}$. For a rigid object in 2D, $s_i, s'_i \in \mathbb{R}^2$, $q_i = (x_i, y_i, \theta_i) \in \mathbb{R}^2 \times SO(2)$; and in 3D, $s_i, s'_i \in \mathbb{R}^3$, q_i

$= (x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i) \in \mathbb{R}^3 \times SO(3)$. The transformations of the skeleton and motion are in the spaces of different dimensions, thus the transformation matrix of the skeleton cannot be directly used to the motion segment. The transformation of a motion segment takes the following steps, see a 2D example in Fig. 13:

- In Fig. 13(a), Sk_{sel} is translated towards Sk_{New} so that their starting points (yellow) overlap. The translating vector is $\overrightarrow{s'_1 s_1}$. Then the position coordinates of every configuration in M_{sel} do the same translation along the vector $\overrightarrow{s'_1 s_1}$, and Sk_{sel} and M_{sel} are updated.
- After initial translation, an angle θ_r between $\overrightarrow{s_1 s_n}$ and $\overrightarrow{s'_1 s_n}$ is got, shown in Fig. 13(b). Then the skeleton Sk_{sel} rotates towards Sk_{New} around its starting point s'_1 by θ_r angle, which makes $\overrightarrow{s'_1 s_n}$ have the same orientation with $\overrightarrow{s_1 s_n}$. The same rotation is done to the position coordinates of every configuration in M_{sel} . The new configuration q_i' in M_{sel} is $q_i' = (x'_i, y'_i, \theta_i)$, shown as the dashed line in Fig. 13(c).
- After the Position Rotation, the configuration q_i' in M_{sel} has the wrong orientation. Then the object placing at q_i' rotates around its center of mass by θ_r , the q_i' becomes $q_i'' = (x'_i, y'_i, \theta_i - \theta_r)$.
- After previous transformations, the skeletons Sk_{sel} and Sk_{New} still mismatch as Fig. 13(c) shows. Given s'_i of Sk_{sel} and its corresponding point s_j of Sk_{New} , the translation vector of s'_i is $\overrightarrow{T_i} = \overrightarrow{s'_i s_j}$. The pair (i, j) is determined by the alignment between Sk_{sel} and Sk_{New} . Every configuration in M_{sel} does the same translation. For example, q_i' translates along $\overrightarrow{T_i}$ to $q_i''' = (x''_i, y''_i, \theta_i - \theta_r)$, where $x''_i = x'_i + \overrightarrow{T_i}(x)$ and $y''_i = y'_i + \overrightarrow{T_i}(y)$.

The transformation cannot guarantee a collision-free motion segment, even though the skeletons of the new scenario segment and the selected scenario segment are very similar. In the implementation, a *Collision Ratio* $q_{\text{collision}}$ is defined as the ratio of $N_{\text{collision}}$ to N_{total} , in which $N_{\text{collision}}$ is the number of in-collision

configurations and N_{total} is the total number of configurations in the transformed motion segment. If the *Collision Ratio* $q_{\text{collision}}$ is larger than a threshold (0.5), this motion segment is considered as a “non-satisfactory” motion; otherwise it is “satisfactory”. If none of the transformed segments are “satisfactory”, the retrieval process will be started on the Failure Tree as previously introduced; otherwise, the user is allowed to select a “satisfactory” motion segment which is close to his preference.

Motion repair: a sparse sampling is made on the new motion segment. In the results of the sampling, the in-collision configurations are retracted to the collision-free configurations. An exact retraction process is the computation of the *Generalized Penetration Depth* which was proposed in [48]. However, the optimality of the generalized penetration depth is not necessary in our application, thus a lower cost algorithm in [25] is adopted for the retraction step. After the retraction, the adjacent configurations are connected by the RRT-Connect [2] algorithm. If all of the connections are successful, a collision-free motion is available in the new scenario segment.

4. Implementation and results

In the following section, our motion planning framework will be applied to a non-convex object moving in a cluttered 2D environment which contains the randomly shaped and located non-convex obstacles. The experimental results demonstrate the efficiency of our method. Moreover, after adequate learning, the dependence on the haptic devices can be greatly reduced in the interaction process by releasing users from the rotation manipulations.

All the experimental results are performed on the software platform Matlab and on a PC with Intel Core i5-3317U, 1.7 GHz, 4G RAM and Intel 4000 GPU.

4.1. User manipulation

For the *interactive motion planning* module in Fig. 2, the method in [25] is adopted, which allows users to manipulate the object with some penetration into the obstacles. The object is manipulated through a joystick and the reference point for the scenario selection is manipulated through a mouse. The distance between two 2D polygons is computed by the method in [49]. Meanwhile, the reference point is not allowed to come up against the obstacles. In this way, users can freely choose the concerned region of workspace. For the Z shaped object in Fig. 14, the accessibility of some scenarios cannot be detected visually. During the interaction, a scenario translating through the environment has been selected by the user, and its skeleton is marked as the dashed curve. Two segments of the scenario (outlined by blue color) are analyzed in the following section. Before the experiments, the learning process has been run in some similar environments and gathered 100 Successful Scenario segments and 75 Failure Scenario segments in the motion library.

4.2. Results in accessible scenarios

The planning results for the first scenario segment are demonstrated in Fig. 15. The sequential search is used for the retrieval process, in which the new scenario segment is compared to each memory scenario segment both in the positive and reverse orders. Here the scenario segments have not been organized into an *R-Tree*, because the sequential search by DTW in 100 scenario segments is fast enough for this application. The ten best retrieval results are sorted by DTW in terms of the radius similarity, and the first five scenario segments are chosen, which are shown in Fig. 15(a). For these five scenario segments, their ID numbers in

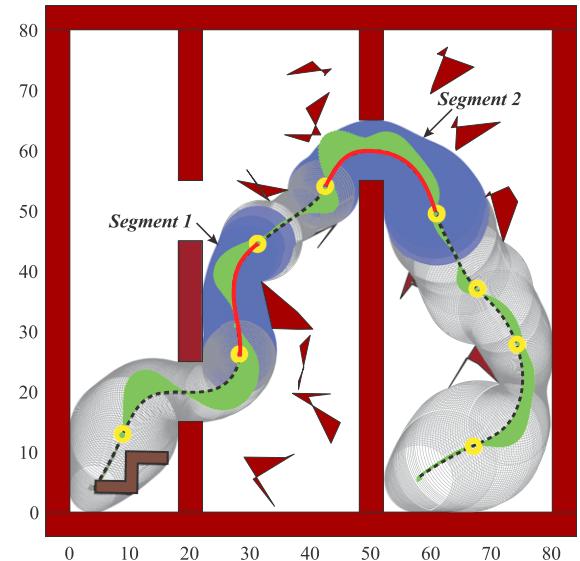


Fig. 14. Two accessible scenario segments selected during the interaction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the motion library and their radius and curvature similarities to the new scenario segment are listed on top of them. The original motions in the five scenario segments are shown in Fig. 15(b).

The motions after transformation are shown in Fig. 15(c), and their *Collision Ratios* are listed below them. The results shown in Fig. 15(a) (b) support our previous hypothesis that the object's motion has a close relationship with the topology of the scenario. The motion always changes greatly on the location where the scenario skeleton has large degree of bending or the sphere radius greatly fluctuates. The same conclusion can be obtained from the results in Fig. 15(c). The collision-free motion is directly transformed from the original motion of the scenario segment “59”, which is the most similar scenario segment to the new scenario segment. It is noteworthy that the original motion in the scenario segment “23” can be transformed to generate a new motion with very small *Collision Ratio*, even though its radius change has a large dissimilarity to that of the new scenario segment. The reason is that, the skeleton of the segment “23” is very similar to that of the new segment, but all the spheres of the segment “23” are smaller than that of the new segment, which amplifies the dissimilarity of the radius but strengthens the relationship between the motion and the scenario skeleton. Since one of the transformed motions is collision-free, further verification is not necessary. The time spent by our method and RRT-connect is shown in Table 1. Our retrieval process has totally made 220 comparisons by DTW with each curve sampled into a sequence of 50 points. If replacing the sequential search with the search on a binary tree, the same retrieving efficiency is available for the motion library containing 2^{50} motion segments. If the binary tree is replaced with the *R-tree*, this number will be further enlarged. Thus, the much more complex applications on the basis of our motion retrieval can be imagined.

The results for the second selected scenario segment are demonstrated in Fig. 16. The search and motion transformation are the same as the first one. The results also support our hypothesis. Similar to the previous example, the 5 retrieved scenario segments with their original motions and transformed motions are shown in Fig. 16(a) (b) (c). The radius changes of both the scenario segments “88” and “61” are similar to the new scenario segment at the same level. However, the skeleton's curvature of the segment “61” is more similar to the new segment than that of the segment “88”, and thus its transformed motion has smaller *Collision Ratio*. The

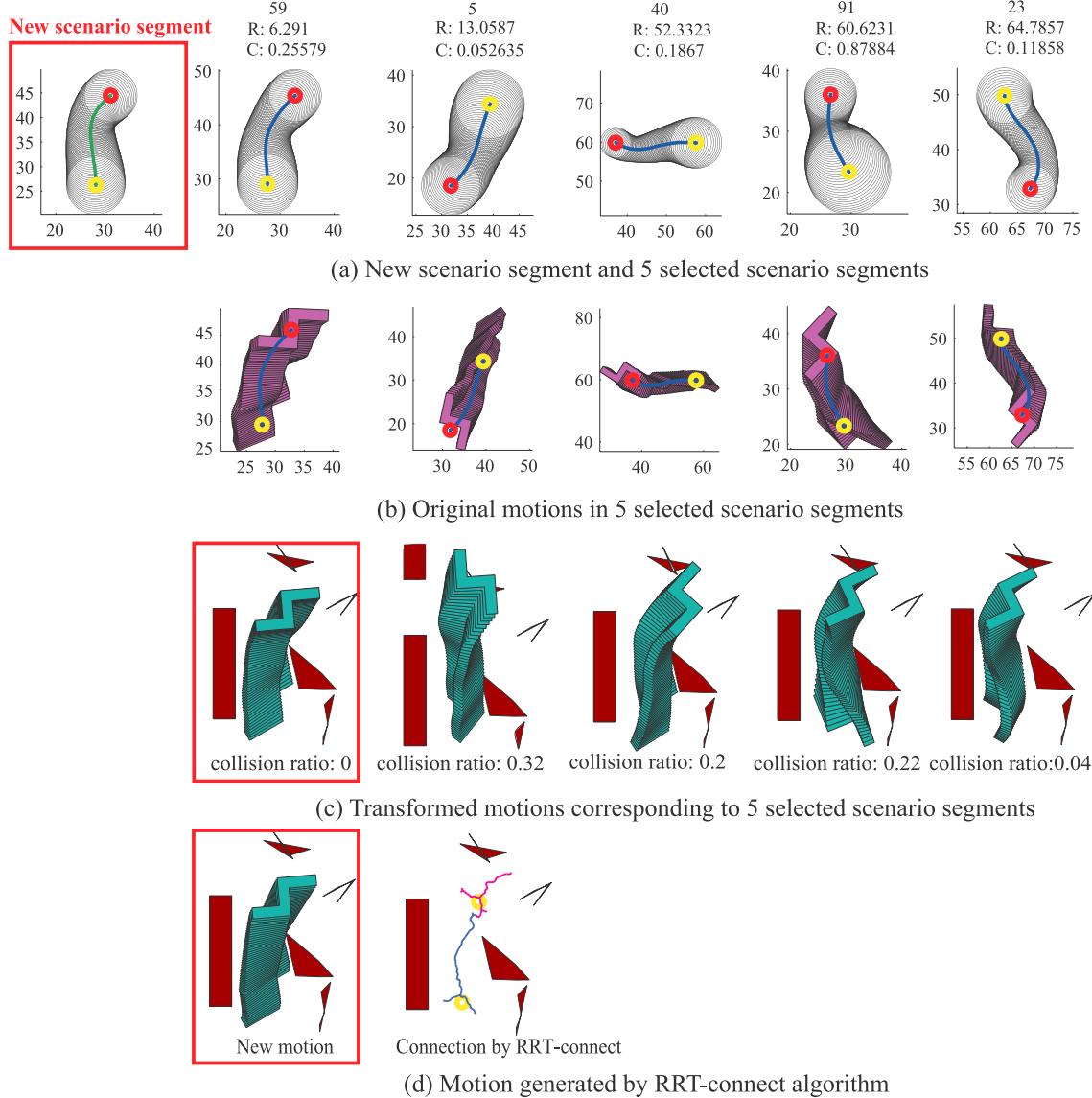


Fig. 15. The motion generation for the first scenario segment and its comparison to RRT-connect.

skeleton's curvature of both the segments "61" and "65" are similar to the new segment at the same level, one of them is in distance value 0.28 and the other is in distance value 0.31. However, compared to the segment "65", the radius change of the segment "61" is much more similar to the new segment, and thus the original motion in the segment "61" is more suitable to generate a new motion with smaller *Collision Ratio*. After the motion transformation, in order to repair the in-collision parts, the transformed motion path is sparsely sampled. If one of the sampled configurations is in-collision, it is retracted to collision-free. After the retraction, every pair of neighboring configurations is connected by RRT-connect. This process (enclosed by a red box in Fig. 16(d)) and its comparison to the connection only by RRT-connect are shown in Fig. 16(d). The results reveal that the transformed motion can offer local guidance to the sampling based motion planners and thus greatly reduce the number of search trials in the configuration space. The time spent by our method and RRT-connect is shown in Table 2.

In the same environment, a complete planning has been carried out, and its results are shown in Fig. 17. In Fig. 17(a), the search trace of RRT-connect is shown and followed by the smoothed motion. In Fig. 17(b), the segmentation result of the selected

Table 1

The time spent by our method and RRT-connect for the first scenario segment.

	T_{total} (s)	$T_{\text{connection}}$ (s)	$T_{\text{retrieval}}$ (s)	$T_{\text{transform}}$ (s)
RRT-connect	3.4874	3.4874	N/A	N/A
Our method	0.3635	N/A	0.2519	0.1116

scenario is shown firstly. The scenario segments are labeled as S_1, S_2, \dots, S_6 . Then the generated motion segments, the search trace and the smoothed motion are demonstrated. In the scenario segments $\{S_1, S_3, S_4, S_5\}$, the transformed motion is collision-free, thus the further search is not necessary. The efficiency of our method is confirmed by the search traces in Fig. 17 and the spent time in Table 3.

4.3. Results in inaccessible scenarios

In the environment of Fig. 14, the accessibility of some scenarios is not explicit, which means it is difficult for users to visually figure out whether the object could pass through the scenario or not. Such an example is shown in Fig. 18, the selected scenario seems possible for the object to pass through but actually it is not. A long

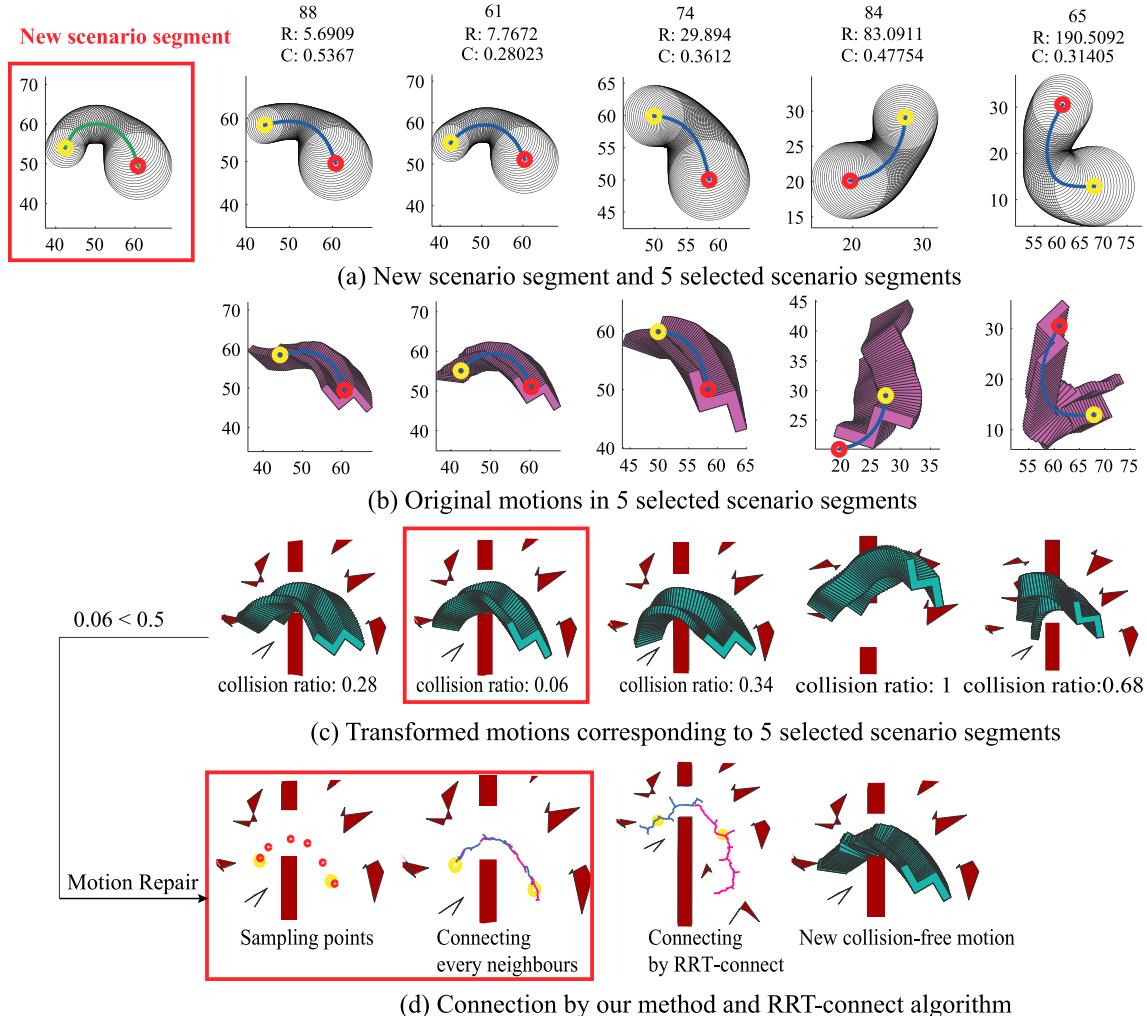


Fig. 16. The motion generation for the second scenario segment and its comparison to RRT-connect. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2

The time spent by our method and RRT-connect for the second scenario segment.

	T_{total} (s)	$T_{\text{retrieval}}$ (s)	$T_{\text{transform}}$ (s)	$\text{Num}_{\text{retraction}}$	$T_{\text{retraction}}$ (s)	$T_{\text{connection}}$ (s)
RRT-connect	8.0725	N/A	N/A	N/A	N/A	8.0725
Our method	1.9533	0.2555	0.1077	0	0.0470	1.5431

narrow section in the middle of the scenario causes the inaccessibility. This problem has its practical value, because the motion path generated by the automatic algorithms cannot guarantee humans' preferences. It is necessary for the designers to quickly check the accessibility of some interested scenarios.

The retrieval has been firstly launched in the Successful Scenario segments. The retrieval and transformation results are shown in Fig. 19(a) (b) (c). In this example, the minimum Collision Ratio is 0.52 which is larger than the threshold 0.5. Thus none of the transformed motions are "satisfactory", meaning that no Successful Scenario segment is considered useful for generating a collision-free motion. In order to verify whether this scenario is accessible or not, the retrieval is started in the Failure Scenario segments. The results are shown in Fig. 19(d). If a Failure Scenario segment is very similar to the new scenario segment, it comes to the conclusion that this new scenario is not possible for the object to pass through. The criterion "very similar" is defined as the similarity distances of the skeleton's curvature and the sphere's radius are smaller than 1 and 10 respectively. The Failure Scenario

segment "26" satisfies the "very similar" criterion, thus the new scenario is considered as infeasible for the object to pass through. The further connection result (enclosed by a red box shown in Fig. 19(e)) verifies our "inaccessible" conclusion. The time spent for the detection of inaccessibility by our retrieval method is shown in the second line of Table 4. Our method only spent 0.73 s to get the "inaccessible" conclusion, and the method in [25] spent 13 s. The same conclusion is not available for RRT-connect, because it has search priority in the large volume region, as illustrated at the end of Fig. 19(e). If limiting the search inside the scenario segment, RRT-connect spends much more time before confirming the inaccessibility of the scenario.

5. Conclusion and extension

Based on the hypothesis that the motion of an object has a close relationship with the topological structure and the volume size of the scenario, we proposed our interactive motion learning and generation framework, and then applied it to a challenging

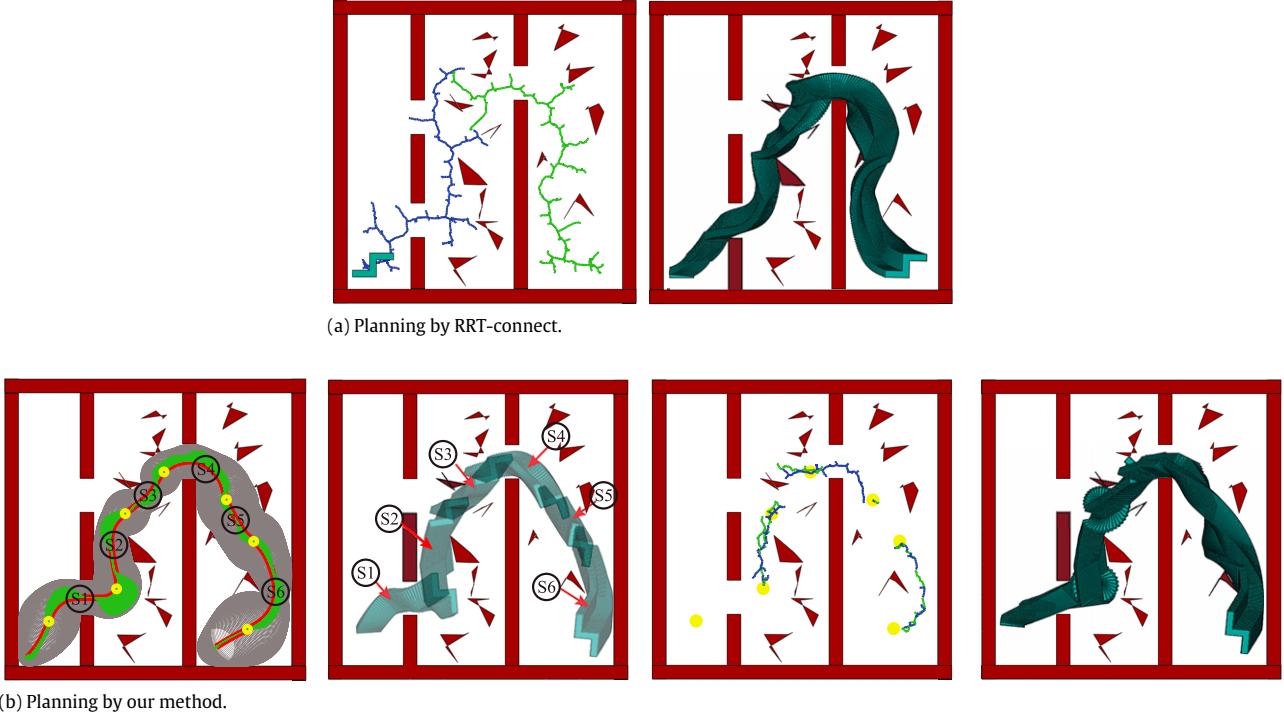


Fig. 17. Comparison between our method and RRT-connect in a complete planning process.

Table 3

The time spent by our method and RRT-connect in a complete planning process.

	T _{total} (s)	T (s)	T _{retrieval} (s)	T _{transform} (s)	T _{retraction} (s)	Num _{retraction}	T _{connection} (s)
RRT-connect	174.243	N/A	N/A	N/A	N/A	N/A	174.243
Our method	8.0376	S1	0.3597	0.2586	0.1011	N/A	N/A
		S2	3.1607	0.2202	0.0739	0.1820	1
		S3	0.2963	0.2225	0.0738	N/A	N/A
		S4	0.3141	0.2223	0.0918	N/A	N/A
		S5	0.3172	0.2202	0.0970	N/A	N/A
		S6	3.5896	0.2344	0.0979	0.1810	1
							3.0763

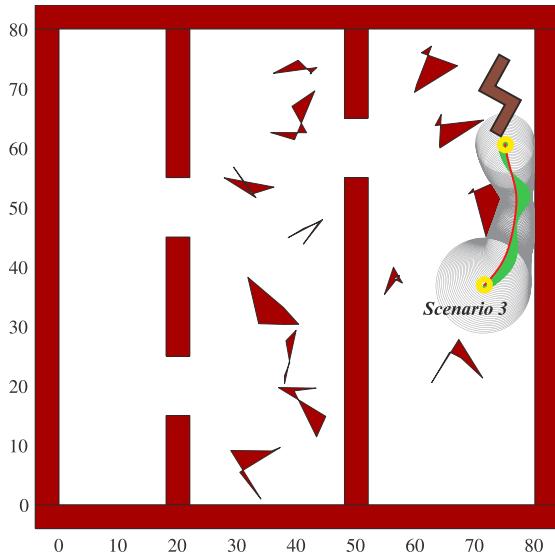


Fig. 18. An inaccessible scenario segment selected during the interaction.

motion planning problem in 2D. For the first time, the topological similarity between scenarios is used to explore the patterns of motion in configuration space. The experimental results verified our hypothesis. Due to the highly efficient scenario retrieval, the

motion library permits recording millions of scenario segments, which is practical for an object moving in a complex environment. Besides, the proposed generation method of the maximal local spheres allows users to freely select concerned scenarios, which releases them from non-convex object's cumbersome rotational manipulations. Hence, this makes the interaction more open for users without expertise. Furthermore, the reduction of the rotational manipulations facilitates the use of natural but non-haptic interactive devices in the complex design verification tasks.

Limitations: For the requirement of the real time interaction, the scenario skeleton is generated by exchanging the accuracy for the efficiency. The inaccuracies can affect the matching result by DTW. Besides, the curvature of the scenario skeleton is translational and rotational invariant, but not invariant in scaling or shearing. When the two compared scenario segments are similar to each other in scaling and shearing, the motion correspondences are not explicit. Moreover, when measuring the similarity between scenarios, how to take the trade-off between the scenario skeleton and the volume size has not been decided.

Extension to 3D: In this paper, we focused on a general motion learning and generation framework which is not limited to a special model. There remains some work to extend the current method to 3D. For example, the generation of the local sphere relies on the distances between the reference point and the nearest obstacle surfaces. The distance can be easily computed in 3D by the search in the point cloud or by the collision detection algorithm. The scenario segmentation and motion transformation

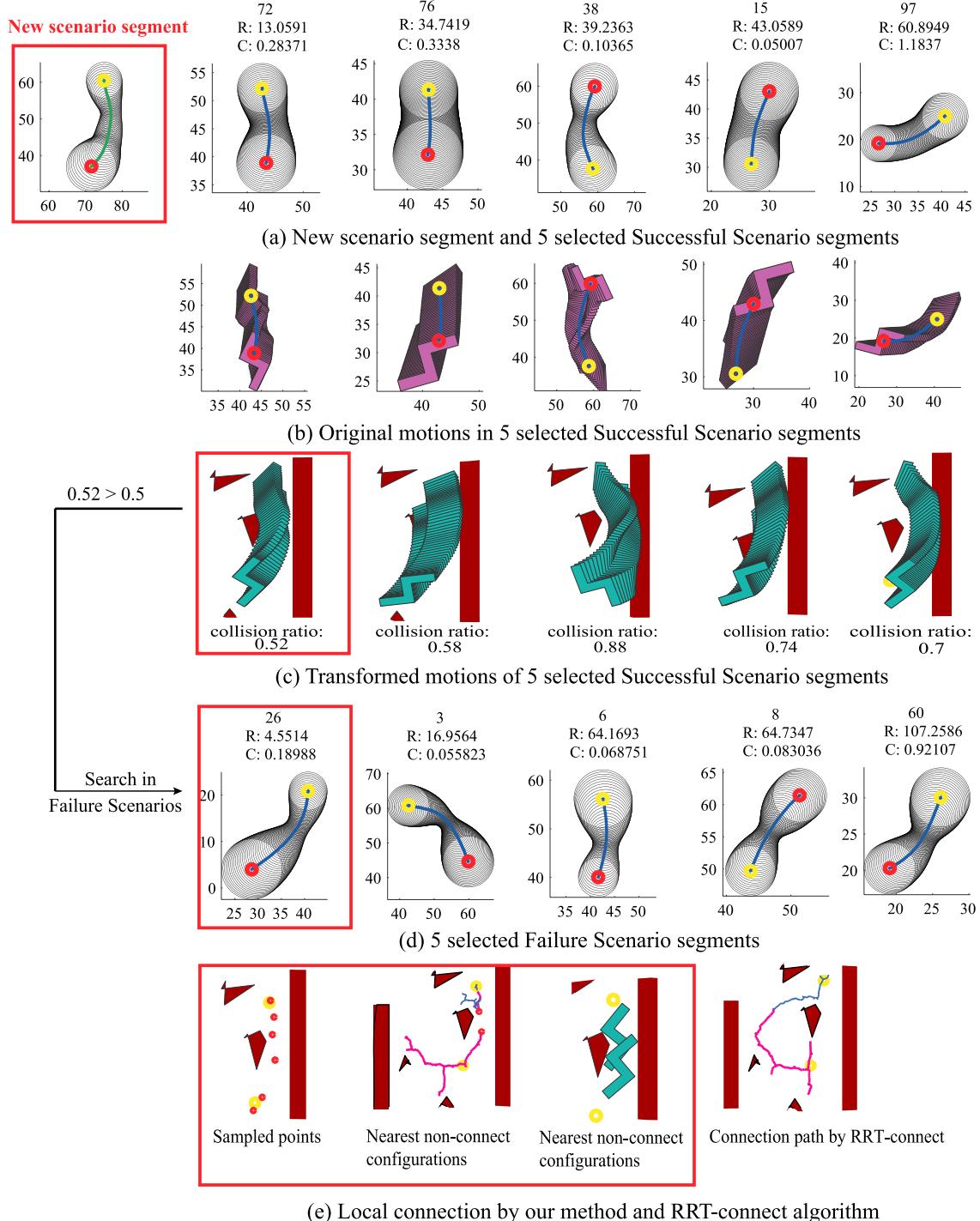


Fig. 19. Accessibility detection for the third scenario segment and its comparison to RRT-connect. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4

The time spent by our method and RRT-connect for the third scenario segment.

	T_{total} (s)	$T_{\text{retrieval}}$ (s)	$T_{\text{transform}}$ (s)	$\text{Num}_{\text{retraction}}$	$T_{\text{retraction}}$ (s)	$T_{\text{connection}}$ (s)
RRT-connect	Fail	N/A				9.4459
Our detection	0.7347	0.3052	0.3181			N/A
Further verification	13.6212	N/A		3	0.5090	13.1122

are independent of the number of dimensions. The curvature of a scenario skeleton in 3D is a space curve. The local cost between the elements of two space curves can be calculated in the same way as

in 2D. Thus the alignment between two scenario skeletons in 3D is also available by DTW. Moreover, the PAA of a scenario skeleton's curvature curve in 3D can be simulated easily in the similar way

as in 2D. Since a user's manipulation degrades to controlling one point in workspace, the interaction in 3D shows no remarkable increase of complexity. In 3D, our method is not suitable to solve the problems with objects hooked together, such as the famous alpha puzzle benchmark.

Other possible extensions: Currently, the method is only applied to the free-flying rigid object, which means its speed and acceleration are not considered. If replacing the planning-from-scratch module with automatic motion planners and taking into account dynamics in the motion transformation, the applications extended to mobile robots can be imagined. For the articulate objects, the correlation between the topological structures of the scenario and the collision-free space $\mathcal{C}_{\text{free}}$ can be also witnessed. However, the affine transformations of motion paths are still not clear. Besides, our current method only considers the topological similarity between scenarios, a more adaptive method in future can integrate with the topological similarity between objects.

References

- [1] Ferré E, Laumond J-P, Arechavaleta G, Estevès C. Progresses in assembly path planning. In: Proceedings of international conference on product lifecycle management. 2005. p. 373–82.
- [2] Kuffner JJ, LaValle SM. RRT-connect: an efficient approach to single-query path planning. In: Proceedings of international conference on robotics and automation. ICRA. 2000. p. 995–1001.
- [3] Morato C, Kaipa KN, Gupta SK. Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters. *Comput-Aided Des* 2013;45:1349–64.
- [4] Zhang L, Manocha D. An efficient retraction-based RRT planner. In: Proceedings of international conference on robotics and automation. ICRA. 2008. p. 3743–50.
- [5] Zhang L, Kim YJ, Manocha D. Efficient cell labelling and path non-existence computation using C-obstacle query. *Int J Robot Res* 2008;27:1246–57.
- [6] Ladeuze N, Fourquet JY, Puel B, Taix M. Haptic assembly and disassembly task assistance using interactive path planning. In: Proceedings of virtual reality conference. VR. 2009. p. 19–25.
- [7] Rosell J, Vázquez C, Pérez A, Iñiguez P. Motion planning for haptic guidance. *J Intell Robot Syst* 2008;53:223–45.
- [8] Ji X, Xiao J. Planning motions compliant to complex contact states. *Int J Robot Res* 2001;20:446–65.
- [9] Tang Peng, Xiao Jing. Automatic generation of high-level contact state space between 3D curved objects. *Int J Robot Res* 2008;27:832–54.
- [10] Hsin-Yi Y, Thomas S, Eppstein D, Amato NM. UOBPRM: a uniformly distributed obstacle-based PRM. In: Proceedings of international conference on intelligent robots and systems. IROS. 2012. p. 2655–62.
- [11] Zheng S, Hsu D, Tingting J, Kurniawati H, Reif JH. Narrow passage sampling for probabilistic roadmap planning. *IEEE Trans Robot* 2005;21:1105–15.
- [12] Redon S, Lin MC. Practical local planning in the contact space. In: Proceedings of international conference on robotics and automation. ICRA. 2005. p. 4200–5.
- [13] Rodriguez S, Xinyu T, Jyh-Ming L, Amato NM. An obstacle-based rapidly-exploring random tree. In: Proceedings of international conference on robotics and automation. ICRA. 2006. p. 895–900.
- [14] Saha M, Latombe J-C, Chang Y-C, Prinz F. Finding narrow passages with probabilistic roadmaps: the small-step retraction method. *Auton Robots* 2005;19:301–19.
- [15] Poirson E, Petiot J-F, Boivin L, Blumenthal D. Eliciting user perceptions using assessment tests based on an interactive genetic algorithm. *J Mech Des* 2013;135:031004.
- [16] Lu W, Petiot J-F. Affective design of products using an audio-based protocol: application to eyeglass frame. *Int J Ind Ergon* 2014;44:383–94.
- [17] Bénabès J, Poirson E, Bennis F. Integrated and interactive method for solving layout optimization problems. *Expert Syst Appl* 2013;40:5796–803.
- [18] He X, Chen Y. Haptic-aided robot path planning based on virtual tele-operation. *Robot Comput-Integr Manuf* 2009;25:792–803.
- [19] Vázquez Hurtado C, Rosell Gratacós J. Haptic guidance based on harmonic functions for the execution of teleoperated assembly tasks. In: Preprints of the IFAC workshop on intelligent assembly and disassembly (IAD). International Federation of Automatic Control. 2008.
- [20] Patil S, Alterovitz R. Interactive motion planning for steerable needles in 3D environments with obstacles. In: 3rd IEEE RAS and EMBS international conference on biomedical robotics and biomechatronics. BioRob. 2010. p. 893–9.
- [21] Taix M, Flavigné D, Ferré E. Human interaction with motion planning algorithm. *J Intell Robot Syst* 2012;67:285–306.
- [22] Pan J, Chitta S, Manocha D. Faster sample-based motion planning using instance-based learning. In: Frazzoli E, Lozano-Perez T, Roy N, Rus D, editors. *Algorithmic foundations of robotics X*. Berlin (Heidelberg): Springer; 2013. p. 381–96.
- [23] Berenson D, Abbeel P, Goldberg K. A robot path planning framework that learns from experience. In: Proceedings of international conference on robotics and automation. ICRA. 2012. p. 3671–8.
- [24] Jetchev N, Toussaint M. Fast motion planning from experience: trajectory prediction for speeding up movement generation. *Auton Robots* 2013;34:111–27.
- [25] Yan Y, Poirson E, Bennis F. Interactive and on-line learning system for assembly task motion planning. In: International design engineering technical conferences and computers and information in engineering conference. Portland (Oregon, USA): ASME; 2013. pp. V004T005A002.
- [26] Hauser K, Bretl T, Harada K, Latombe J-C. Using motion primitives in probabilistic sample-based planning for humanoid Robots. In: Akella S, Amato N, Huang W, Mishra B, editors. *Algorithmic foundation of robotics VII*. Berlin (Heidelberg): Springer; 2008. p. 507–22.
- [27] Kallmann M, Bargmann R, Mataric M. Planning the sequencing of movement primitives. In: Proceedings of the international conference on simulation of adaptive behavior. SAB. 2004. p. 193–200.
- [28] Pastor P, Hoffmann H, Asfour T, Schaal S. Learning and generalization of motor skills by learning from demonstration. In: Proceedings of international conference on robotics and automation. ICRA. 2009. p. 763–8.
- [29] Ma J, Bae SW, Choi S. 3D medial axis point approximation using nearest neighbors and the normal field. *Vis Comput* 2012;28:7–19.
- [30] Geraerts R, Overmars MH. The corridor map method: real-time high-quality path planning. In: Proceedings of international conference on robotics and automation. ICRA. 2007. p. 1023–8.
- [31] Yeh H-YC, Denny J, Lindsey A, Thomas S, Amato NM. Uniformly sampling the medial axis. Technical report. Parasol Laboratory, Department of Computer Science, Texas A&M University. 2013.
- [32] Lien J-M, Keyser J, Amato NM. Simultaneous shape decomposition and skeletonization. In: Proceedings of the ACM symposium on solid and physical modeling. Cardiff (Wales, United Kingdom): ACM; 2006. p. 219–28.
- [33] Kustra J, Jalba A, Telea A. Probabilistic view-based 3D curve skeleton computation on the GPU. In: Proceedings of 8th international joint conference on computer vision, imaging and computer graphics theory and applications. 2013. p. 237–46.
- [34] Cornea ND, Silver D, Min P. Curve-skeleton properties, applications, and algorithms. *IEEE Trans Vis Comput Graphics* 2007;13:530–48.
- [35] Jiang W, Xu K, Cheng Z-Q, Martin RR, Dang G. Curve skeleton extraction by coupled graph contraction and surface clustering. *Graph Models* 2013;75:137–48.
- [36] Kin-Chung Au O, Tai CL, Cohen-Or D, Zheng Y, Fu H. Electors voting for fast automatic shape correspondence. In: Computer graphics forum. Wiley Online Library; 2010. p. 645–54.
- [37] Agrawal R, Faloutsos C, Swami A. Efficient similarity search in sequence databases. In: Lomet D, editor. *Foundations of data organization and algorithms*. Berlin (Heidelberg): Springer; 1993. p. 69–84.
- [38] Müller M. Information retrieval for music and motion. Springer; 2007.
- [39] Cui M, Femiani J, Hu J, Wonka P, Razdan A. Curve matching for open 2D curves. *Pattern Recognit Lett* 2009;30:1–10.
- [40] Pelekis N, Kopanakis I, Marketos G, Ntoutsios I, Andrienko G, Theodoridis Y. Similarity search in trajectory databases. In: Proceedings of the 14th international symposium on temporal representation and reasoning. IEEE Computer Society; 2007. p. 129–40.
- [41] Berndt DJ, Clifford J. Finding patterns in time series: a dynamic programming approach. In: Usama MF, Gregory P-S, Padhraic S, Ramasamy U, editors. *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence; 1996. p. 229–48.
- [42] Bollobás B, Das G, Gunopulos D, Mannila H. Time-series similarity problems and well-separated geometric sets. In: Proceedings of the annual symposium on computational geometry. Nice (France): ACM; 1997. p. 454–6.
- [43] Chen L, Özsu MT, Oria V. Robust and fast similarity search for moving object trajectories. In: Proceedings of the SIGMOD international conference on management of data. Baltimore (Maryland): ACM; 2005. p. 491–502.
- [44] Keogh E, Ratanamahatana CA. Exact indexing of dynamic time warping. *Knowl Inf Syst* 2005;7:358–86.
- [45] Nelaturi S, Shapiro V. Solving inverse configuration space problems by adaptive sampling. *Comput-Aided Des* 2013;45:373–82.
- [46] Varadhan G, Kim YJ, Krishnan S, Manocha D. Topology preserving approximation of free configuration space. In: Proceedings of international conference on robotics and automation. ICRA. 2006. p. 3041–8.
- [47] Kroon D-J. 2D line curvature and normals. In: Matlab file exchange. 2011. <http://www.mathworks.fr/matlabcentral/fileexchange/32696-2d-line-curvature-and-normals>.
- [48] Zhang L, Kim YJ, Varadhan G, Manocha D. Generalized penetration depth computation. *Comput-Aided Des* 2007;39:625–38.
- [49] Jacquenot G. Minimum distance between two polygons. In: Matlab file exchange. 2008. <http://www.mathworks.fr/matlabcentral/fileexchange/22444-minimum-distance-between-two-polygons>.