



**HAL**  
open science

# Interactive and Learning-Based Motion Planning Method for the Accessibility Verification of the Product Design

Yu Yan

► **To cite this version:**

Yu Yan. Interactive and Learning-Based Motion Planning Method for the Accessibility Verification of the Product Design. Engineering Sciences [physics]. Ecole Centrale de Nantes (ECN), 2014. English. NNT: . tel-01174198

**HAL Id: tel-01174198**

**<https://hal.science/tel-01174198>**

Submitted on 8 Jul 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Thèse de Doctorat

Yu YAN

*Mémoire présenté en vue de l'obtention du  
grade de Docteur de l'Ecole Centrale de Nantes  
sous le label de L'Université Nantes Angers Le Mans*

**École doctorale :** *Sciences Pour l'Ingénieur, Géosciences, Architecture*

**Discipline :** *Génie Mécanique, Productique, Transport*

**Unité de recherche :** *Institut de Recherche en Communications et Cybernétique de Nantes*

**Soutenu le 09/12/2014**

## Interactive and Learning-Based Motion Planning Method for the Accessibility Verification of the Product Design

### JURY

Président :	<b>Jean-Yves FOURQUET</b> , Professeur des universités, Ecole Nationale d'Ingénieurs de Tarbes
Rapporteurs :	<b>Bernard YANNOU</b> , Professeur des universités, Laboratoire Génie Industriel, Ecole Centrale Paris <b>Xavier FISHER</b> , Professeur des universités, École Supérieure Des Technologies Industrielles Avancées
Examineurs :	<b>Jean-Yves FOURQUET</b> , Professeur des universités, Ecole Nationale d'Ingénieurs de Tarbes <b>Yannick RAVAUT</b> , Ingénieur, Thales communications & security <b>Fouad BENNIS</b> , Professeur des universités, IRCCyN, Ecole Centrale de Nantes <b>Emilie POIRSON</b> , Maître de Conférences, IRCCyN, Ecole Centrale de Nantes
Directeur de Thèse :	<b>Fouad BENNIS</b> , Professeur des universités, IRCCyN, Ecole Centrale de Nantes
Co-encadrant de Thèse :	<b>Emilie POIRSON</b> , Maître de Conférences, IRCCyN, Ecole Centrale de Nantes

# Thèse de Doctorat

Yu YAN

## Méthode interactive et par l'apprentissage pour la generation de trajectoire en conception du produit

Interactive and Learning-Based Motion Planning Method for the Accessibility Verification of the Product Design

### Résumé

L'accessibilité est un facteur important pris en compte dans la validation et la vérification en phase de conception du produit et augmente généralement le temps et les coûts de cette phase. Ce domaine de recherche a eu un regain d'intérêt ces quinze dernières années avec notamment de nouveaux planificateurs de mouvement. Cependant, les performances de ces méthodes sont généralement très faibles lorsque le problème se caractérise par des passages étroits des assemblages complexes composés d'un grand nombre de pièces. Cela conduit souvent à des scènes à forte densité d'obstacles. Malheureusement, les manipulations manuelles des humains dans le passage étroit montrent toujours beaucoup de difficultés en raison des limitations des dispositifs interactifs ou la capacité cognitive. Pendant ce temps, les défis de l'analyse de la réponse finale des utilisateurs dans le processus de conception promeut l'intégration avec la participation directe des concepteurs.

Afin d'accélérer la planification dans le passage étroit et trouver le chemin le plus conforme aux préférences de l'utilisateur, une nouvelle méthode de planification de mouvement interactif est proposée. Nous avons souligné la performance de notre algorithme dans certains scénarios difficiles en 2D et 3D environnement.

Ensuite, une hypothèse est proposée sur la corrélation entre la structure topologique du scénario et la trajectoire dans le passage étroit. La méthode basée sur les courbures est utilisée pour explorer cette corrélation et un cadre de planification de mouvement interactif qui peut apprendre de l'expérience est construit dans cette thèse. Nous soulignons la performance de notre cadre sur un problème difficile en 2D, dans lequel un objet non-convexe passe à travers un environnement encombré rempli d'obstacles non-convexes de forme aléatoire et situés.

### Mots clés

**vérification de l'accessibilité, de la planification de chemin, interaction homme-machine, agencement, passage étroit, récupération de scénario, la conception des produits**

### Abstract

The accessibility is an important factor considered in the validation and verification phase of the product design and usually dominates the time and costs in this phase. Defining the accessibility verification as the motion planning problem, the sampling based motion planners gained success in the past fifteen years. However, the performances of them are usually shackled by the narrow passage problem arising when complex assemblies are composed of large number of parts, which often leads to scenes with high obstacle densities. Unfortunately, humans' manual manipulations in the narrow passage always show much more difficulties due to the limitations of the interactive devices or the cognitive ability. Meanwhile, the challenges of analyzing the end users' response in the design process promote the integration with the direct participation of designers.

In order to accelerate the path planning in the narrow passage and find the path complying with user's preferences, a novel interactive motion planning method is proposed. In this method, the integration with a random retraction process helps reduce the difficulty of manual manipulations in the complex assembly/disassembly tasks and provide local guidance to the sampling based planners. Then a hypothesis is proposed about the correlation between the topological structure of the scenario and the motion path in the narrow passage. The topological structure refers to the medial axis (2D) and curve skeleton (3D) with branches pruned. The correlation runs in an opposite manner to the sampling based method and provide a new perspective to solve the narrow passage problem. The curve matching method is used to explore this correlation and an interactive motion planning framework that can learn from experience is constructed in this thesis. 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.

### Key Words

**accessibility verification, path planning, human computer interaction, layout design problem, narrow passage, scenario retrieval, product design**

# Acknowledgements

First and foremost, I would like to express my deepest gratitude to my thesis director, Prof. Fouad BENNIS, for his caring, patience, and providing me with an excellent atmosphere for doing research. The good advice, support and friendship of my second supervisor, Dr. Emilie POIRSON, has been invaluable on both an academic and a personal level, for which I am extremely grateful. Their important guidance and continuing encouragements have allowed me to make much progress in my studies over the multidisciplinary domains.

My greetings go also to Prof. Bernard YANNOU and Prof. Xavier FISHER, who honored me by accepting to be reviewers of my thesis. I also thank Prof. Jean-Yves FOURQUET and Dr. Yannick RAVAUT for taking part of the jury. All their comments and suggestions are very useful.

I would like to thank the China Scholarship Council for providing me with the scholarship during my three years at Ecole Centrale de Nantes, without this financial support I could not have undertaken this study.

Many thanks also to my colleagues at L'Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), especially the members of Team IS3P. The study and discussion in IS3P have brought me into contact with the cutting edge topics about the layout design, interactive design, affective design, robust design and the additive manufacturing.

I am indebted to my many friends for providing a stimulating and fun filled environment in Ecole Centrale de Nantes. I am thankful to Julien BÉNABÈS, Roland RENIER, Ruina MA, Weihua LU, Yicha ZHANG, Weijun Wang, Fang LIU, Marie HOARAU, Philip LONG, Jie ZHAO, Jian HAN, Gang LI, Zheng LI, Yingjing LIU, Guanqin MA, Fan ZHANG, Mi ZHANG, Mayao CHENG, Qian ZHAO, Mohamed Anis DHUIEB, Poiree RAPHAEL.

I would also like to thank my family, especially my parents who always supported me with their best wishes. Thank them for all their love and encouragement. Most of all, I would like to thank my girlfriend Yang CHEN for her endless love and support.

---

---

*“The opposite of a fact is a falsehood, but the opposite of one  
profound truth may well be another profound truth.”*

Niels Bohr (1885-1962)  
Danish physicist

---

# Abstract

The accessibility is an important factor considered in the validation and verification phase of the product design, and usually dominates the time and costs in this phase. Defining the accessibility verification as the motion planning problem, the sampling based motion planners gained success in the past fifteen years. However, the performances of them are usually degraded by the narrow passage problem arising when complex assemblies are composed of large number of parts, which often leads to scenes with high obstacle densities. Unfortunately, humans' manual manipulations in the narrow passage always show much more difficulties due to the limitations of the interactive devices or the cognitive ability. Meanwhile, the challenges of analyzing the end users' response in the design process promote the integration with the direct participation of designers.

In order to accelerate the path planning in the narrow passage and find the path complying with user's preferences, a novel interactive motion planning method is proposed. In this method, the integration with a random retraction process helps reduce the difficulty of manual manipulations in the complex assembly/disassembly tasks and provide local guidance to the sampling based planners. We highlighted the performance of our algorithm in some difficult scenarios in 2D and 3D environment.

Then a hypothesis is proposed about the correlation between the topological structure of the scenario and the motion path in the configuration space. The topological structure refers to the medial axis (2D) and curve skeleton (3D) with branches pruned. The correlation runs in an opposite manner to the sampling based method and provide a new perspective to solve the narrow passage problem. The curve matching method is used to explore this correlation and an interactive motion planning framework that can learn from experience is constructed in this thesis. Our framework provides some good facilities, such as governing the motion experience with the R-tree, efficiently retrieving motion segments from a large database, relieving users from burdensome rotational manipulations and determining the inaccessibility in a belief range without a complete planning. 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.

**Keywords:** accessibility verification, path planning, human computer interaction, narrow passage, scenario retrieval, product design



# Résumé

L'accessibilité est un facteur important pris en compte dans la validation et la vérification en phase de conception du produit et augmente généralement le temps et les coûts de cette phase. Ce domaine de recherche a eu un regain d'intérêt ces quinze dernières années avec notamment de nouveaux planificateurs de mouvement. Cependant, les performances de ces méthodes sont généralement très faibles lorsque le problème se caractérise par des passages étroits des assemblages complexes composés d'un grand nombre de pièces. Cela conduit souvent à des scènes à forte densité d'obstacles. Malheureusement, les manipulations manuelles des humains dans le passage étroit montrent toujours beaucoup de difficultés en raison des limitations des dispositifs interactifs ou la capacité cognitive. Pendant ce temps, les défis de l'analyse de la réponse finale des utilisateurs dans le processus de conception promettent l'intégration avec la participation directe des concepteurs.

Afin d'accéder à la planification dans le passage étroit et trouver le chemin le plus conforme aux préférences de l'utilisateur, une nouvelle méthode de planification de mouvement interactif est proposée. Dans cette méthode, l'intégration d'un processus de rétraction aléatoire permet de réduire la difficulté de manipulations manuelles dans les tâches de montage / démontage complexes et fournir des conseils locaux pour les planificateurs d'échantillonnage sur la base. Nous avons souligné la performance de notre algorithme dans certains scénarios difficiles en 2D et 3D environnement.

Ensuite, une hypothèse est proposée sur la corrélation entre la structure topologique du scénario et la trajectoire dans le passage étroit. La structure topologique se réfère à l'axe médian (2D) et la courbe squelette (3D) avec des branches taillées. La corrélation fonctionne d'une manière opposée à la méthode d'échantillonnage sur la base et fournit une nouvelle perspective pour résoudre le problème de passage étroit. La méthode basée sur les courbures est utilisée pour explorer cette corrélation et un cadre de planification de mouvement interactif qui peut apprendre de l'expérience est construit dans cette thèse. Nous soulignons la performance de notre cadre sur un problème difficile en 2D, dans lequel un objet non-convexe passe à travers un environnement encombré rempli d'obstacles non-convexes de forme aléatoire et situés.

**Mots-clés:** vérification de l'accessibilité, de la planification de chemin, interaction homme-machine, passage étroit, récupération de scénario, la conception des produits



# List of Figures

Figure 1. Our interactive motion planning framework and the chapter structure .....	2
Figure 1.1. Accessibility verification in the product lifecycle .....	6
Figure 1.2. Configuration space of a 2D translating object .....	9
Figure 1.3. Configuration space of an object freely moving in 2D plane .....	10
Figure 1.4. Configuration of a free-flying rigid object in 3D.....	10
Figure 1.5. Some models of haptic devices.....	16
Figure 1.6. Advantage and disadvantage of human’s capability of motion planning .....	18
Figure 1.7. The captured frames of a dexterous shooting action of a football player (Courtesy of <a href="http://axeetech.com/">http://axeetech.com/</a> ) .....	19
Figure 2.1. Loop integrating designers into the motion planning process .....	26
Figure 2.2. Comparison of retraction processes in different methods.....	31
Figure 2.3. The cases related to the three rules considered in our retraction design.....	32
Figure 2.4. Penetration estimation in 2D and 3D.....	35
Figure 2.5. The construction process of spheres array .....	36
Figure 2.6. Sphere array in different resolution .....	36
Figure 2.7. Two cases with invalid penetration .....	37
Figure 2.8. The situations of convergence in two different cases .....	38
Figure 2.9. The comparison between a ‘good’ situation and a ‘bad’ situation .....	39
Figure 2.10. Experiment results of random retraction algorithm in the first case.....	41
Figure 2.11. Experiment results of random retraction algorithm in the second case .....	42
Figure 2.12. Penetration volume in different resolution .....	43
Figure 2.13. Valid and invalid retraction cases .....	44
Figure 2.14. The comparison of retraction time and number of retraction steps in different resolution and step length .....	46
Figure 2.15. Interactive motion planning system.....	47
Figure 2.16. Interactive planning in successful case and failed case .....	48
Figure 2.17. Connection of seeds by RRT-connect algorithm .....	50
Figure 2.18. Successful case for an L shape part passing through a narrow passage .....	51
Figure 2.19. Exploration trace of basic BiRRT.....	51
Figure 2.20. Comparison between basic BiRRT and our interactive process .....	52
Figure 2.21. Failed case for an L shape part passing through a narrow passage.....	54
Figure 2.22. Simple case for a z shape part passing through narrow passages .....	54

Figure 2.23. The Planning process by our method in different sampling rates.....	56
Figure 3.1. Motion segment learned in a successful planning task.....	61
Figure 3.2. Learned motion segment is applied to the four similar scenarios.....	61
Figure 3.3. Similar scenario segments and their topological representations .....	63
Figure 3.4. Overview of the motion learning process .....	64
Figure 3.5. Sphere generation according to the position of the reference point.....	66
Figure 3.6. Sphere generation in a planar cluttered environment in two cases .....	67
Figure 3.7. Segmentation of scenario.....	69
Figure 3.8. Calculation of scenario skeleton's curvature .....	71
Figure 3.9. Alignment between two similar skeleton segments based on curvature.....	72
Figure 3.10. Imposing global constraint to limit the warping path .....	75
Figure 3.11. Sorted matching results for 4 query scenario skeleton segments by DTW using curvature feature .....	76
Figure 3.12. The <i>Upper</i> and <i>Lower</i> constraints on the curvature curve of a scenario skeleton .....	79
Figure 3.13. Piecewise aggregate approximation (PAA) of bounding constraints and the distance calculation between a query scenario segment and a minimum bounding rectangle. ....	80
Figure 3.14. An example of R-tree (Guttman 1984) .....	81
Figure 3.15. Bad and good split of an R-tree (Guttman 1984).....	82
Figure 3.16. Process clustering scenario skeleton and its motion into motion library .....	83
Figure 3.17. Good and bad split of an R-tree in motion library .....	85
Figure 3.18. An R-tree constructed from ten scenario segments .....	86
Figure 3.19. Searching time in motion library with the R-tree structure .....	87
Figure 4.1. Overview of motion generation process .....	90
Figure 4.2. Motion Generation Process.....	91
Figure 4.3. Two invalid situations during the generation of the controlled point's trajectory .....	93
Figure 4.4. Modified PRM is used to generate the trace of the controlled point .....	93
Figure 4.5. The computation of the length of roadmap edge .....	94
Figure 4.6. Scenario retrieval process .....	95
Figure 4.7. The ten retrieved scenario segments with similar skeleton to the query scenario segment (ID is the identifier number in the motion library; C is the dissimilarity distance of the curvature of scenario skeleton).....	96
Figure 4.8. The five scenario segments selected from the ten retrieved segments (ID is the identifier number in the motion library; R is the dissimilarity distance of the radius of scenario).....	96

---

Figure 4.9. Motion transformation according to the skeleton transformation.....	97
Figure 4.10. Two feasible scenario segments selected in an interaction.....	99
Figure 4.11. The motion generation for the first selected scenario segment and its comparison to RRT-connect.....	101
Figure 4.12. The motion generation for the second selected scenario segment and its comparison to RRT-connect.....	102
Figure 4.13. Comparison between our method and RRT-connect in a complete planning process	103
Figure 4.14. A non-feasible scenario segment selected during the interaction .....	104
Figure 4.15. Feasibility detecting for the third selected scenario segment and its comparison to RRT-connect.....	105
Figure 4.16. Two articulated models (The first is freely moveable in the workspace; the second is nailed to one point in the workspace) .....	106
Figure 4.17. Examples of reusing motion segments of two articulated model .....	107
Figure 4.18. The comparison of the $\mathcal{CS}_{free}$ of three different scenarios with $\mathcal{CS} = \mathbb{R} \times \mathbb{T}^2$ ...	108
Figure 4.19. The comparison of the $\mathcal{CS}_{free}$ of three different scenarios with $\mathcal{CS} = \mathbb{R} \times \mathbb{T}$ .....	109
Figure 4.20. The model of a simple car.....	110
Figure 4.21. The planning results of the path of Dubin's car by RRT algorithm in three tightly constrained scenarios with different topological structures.....	112
Figure 4.22. The Accessibility verification of a forklift in an industrial case with the construction of the scenario.....	113



# List of Tables

Table 1.1. The comparison between different interactive devices .....	17
Table 2.1. Comparison between different interactive motion planning algorithm.....	29
Table 2.2. The spent time and the change of $\delta$ in the first retraction case .....	41
Table 2.3. The spent time and the change of $\delta$ in the second retraction case .....	42
Table 2.4. The spent time and rates of successful retraction in different resolution and step length	45
Table 2.5. The number of retraction steps in different resolution and step length.....	45
Table 2.6. The spent time of penetration evaluation in different resolution .....	45
Table 2.7. Comparison between BiRRT and our interactive process in case of radius = 5 .....	52
Table 2.8. Comparison between BiRRT and our interactive process in case of different radius...	52
Table 2.9. Planning result by RRT-connect .....	56
Table 2.10. Planning result by our method in the sampling rate 5.....	57
Table 2.11. Planning result by our method in the sampling rate 10.....	57
Table 2.12. Planning result by our method in the sampling rate 15.....	57
Table 4.1. Comparison between our method and RRT-connect on the motion generation in the first example .....	101
Table 4.2. Comparison between our method and RRT-connect on the motion generation in the second example.....	102
Table 4.3. Comparison between our method and RRT-connect in a complete planning process	103
Table 4.4. Comparison between our method and RRT-connect on the motion generation in the third example .....	106



# Contents

<b>Acknowledgements</b> .....	<b>i</b>
<b>Abstract</b> .....	<b>i</b>
<b>Résumé</b> .....	<b>iii</b>
<b>List of Figures</b> .....	<b>v</b>
<b>List of Tables</b> .....	<b>ix</b>
<b>Contents</b> .....	<b>xi</b>
<b>Introduction</b> .....	<b>1</b>
<b>1 Problem Introduction and Literature Review</b> .....	<b>5</b>
1.1. Accessibility verification in the product design .....	5
1.1.1. Problem formulation .....	8
1.1.2. Configuration space .....	9
1.2. Accessibility verification by automatic algorithm.....	11
1.2.1. Sampling based motion planners.....	11
1.2.2. Narrow passage problem.....	12
1.2.3. Complete planning .....	13
1.2.3. Ergonomics analysis.....	13
1.3. Accessibility verification by manual manipulation .....	14
1.3.1. Manual manipulation in virtual environment with interactive devices .....	14
1.3.2. Constraints for humans .....	17
1.4. Introduction to the motion learning .....	18
1.4.1. Learning in the configuration space .....	20
1.4.2. Learning in the workspace .....	22
<b>2 Interactive Motion Planning</b> .....	<b>25</b>
2.1. Introduction .....	25
2.2. Previous work in the interactive motion planning (IMP) .....	27
2.3. Retraction based on the penetration depth .....	29
2.3.1. Retraction based motion planner.....	30
2.3.2. Random retraction method .....	32
2.3.3. Estimation of the penetration .....	34
2.3.4. Convergence.....	37

2.3.5. Retraction results in applications .....	39
2.3.6. Conclusion.....	46
2.4. IMP based on the retraction and the local connection .....	46
2.4.1. System global view .....	47
2.4.2. Seeds connection by the RRT-connect algorithm .....	49
2.4.3. Application cases .....	50
2.5. Conclusion.....	57
<b>3 Motion Learning Based on Scenario Matching.....</b>	<b>59</b>
3.1. Introduction .....	59
3.2. The hypothesis from an observation .....	60
3.3. Overview of the motion learning process .....	64
3.4. Scenario description and scenario skeleton generation .....	65
3.4.1. Medial axis .....	65
3.4.2. Curve skeleton.....	66
3.4.3. Scenario skeleton generation.....	66
3.4.4. Scenario segmentation .....	68
3.5. Similarity measurement of scenarios by DTW .....	69
3.5.1. Feature selection on scenario skeleton .....	70
3.5.2. Similarity measurement of scenarios by Dynamic Time Warping (DTW).....	71
3.5.3. Conclusion.....	77
3.6. Organization of scenario segments in motion library .....	77
3.6.1. Indexing structure of scenario segments .....	78
3.6.2. R-tree introduction .....	80
3.6.3. Clustering scenario segments with R-tree .....	82
3.7. Conclusion.....	87
<b>4 Interactive Motion Planning with Learned Experience.....</b>	<b>89</b>
4.1. Introduction .....	89
4.2. Motion generation process .....	90
4.2.1. Scenario selection.....	91
4.2.2. Motion retrieval.....	95
4.2.3. Motion transformation .....	97
4.2.4. Motion repair.....	98
4.3. Application results .....	98
4.3.1. Results in feasible scenarios.....	99
4.3.2. Results in non-feasible scenarios .....	104

---

4.4. Extension.....	106
4.4.1. Extension to the motion planning of articulated object.....	106
4.4.2. Extension to the differential motion planning.....	110
4.5. Conclusion.....	114
<b>5 General Conclusions and perspectives.....</b>	<b>117</b>
<b>Bibliography .....</b>	<b>119</b>
<b>Personal Publications.....</b>	<b>132</b>



# Introduction

The accessibility verification plays an important role in the process of product design, which usually dominates the cost and time of the design circle. The accessibility can be verified by the automatic algorithms or the manual manipulations equipped with the interactive devices. Defining the accessibility verification as the motion planning problem, the sampling based algorithms gained success on solving this kind of problem in the past fifteen years. However, all of them have some level of difficulty if the object's free space has narrow passages which are regularly seen in some complex virtual assembly/disassembly tasks. Moreover, considering the evaluation of the constraints to the articulate system of human skeleton, and the preferences of users, the automatic algorithms always suffer from the redundant degrees of freedom and the difficulty of modeling users' preferences. The accessibility verification also promotes challenges to the manual manipulation of humans. Although humans possess advantages in the ability of global motion planning, they can lose their power in the local precise operations. Thus the manual manipulation is time-consuming for some tasks in which there is small tolerance between the manipulated object and the obstacles, or poor visibility. Besides, haptic feedback plays an important role in complex virtual manipulations. However, the haptic devices introduce limitations to the interaction as well, such as high cost, low degrees of freedom, and cumbersome mechanical structure, which usually prevent the integration with natural non-haptic devices in some tasks. All of these problems promote the interaction between users and automatic path planners in the accessibility verification process.

## Summary of contributions

In this thesis, the objective is to improve the accessibility verification process for the product design. In considering the limitations in the automatic motion planning algorithms and human's manipulation abilities, our research concentrates on integrating the algorithms and humans together, for the purposes such as accelerating the verification, reducing the difficulty of virtual manipulation and finding the motion path complying with the users' preferences. Specially, the contributions include:

- A novel random retraction algorithm is proposed to retract the penetrated configuration created by user to the configuration without penetration. Based on the retraction algorithm, a novel retraction based interactive motion planning method is proposed. Although the concept of some components in this method have been proposed by some previous studies, the integration of them with our random retraction method is new.

- A new finding about the correlation between the topological information of scenarios and the motion path is studied. Then this correlation has been explored to establish a motion learning paradigm.
- Scenario representation method is proposed, with which the similarity of scenarios is measured by the curve matching method.
- The efficient motion segments clustering and retrieval methods are established with the R-tree structure.
- The retraction based interactive motion planning method and motion learning method have been integrated as a new superior interactive motion planning framework which can learn from experience and release user from burdensome manipulations.

## Organization of the thesis

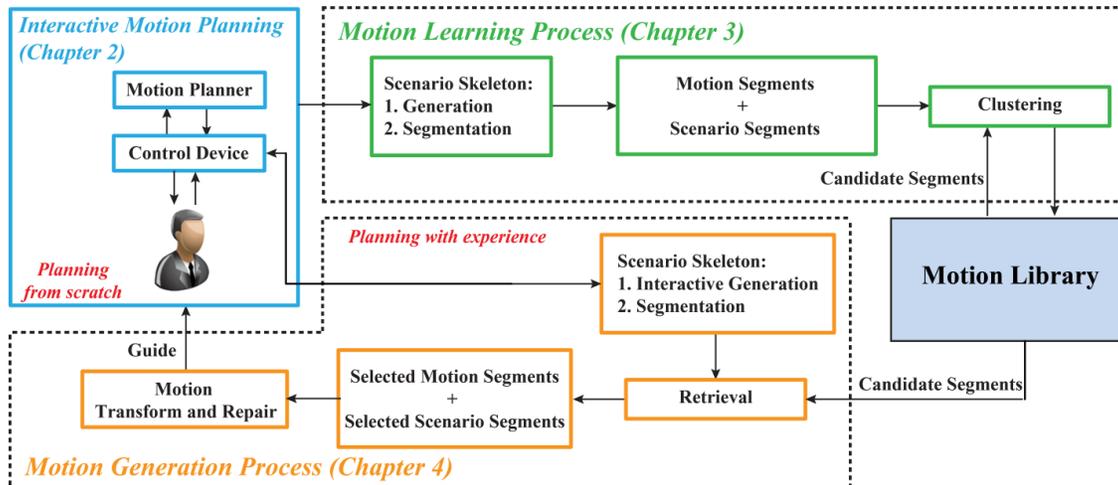


Figure 1. Our interactive motion planning framework and the chapter structure

A global view to our interactive motion planning framework is demonstrated in Fig. 1. Without considering motion learning, the motion is planned from scratch by the retraction based interactive motion planning module which is discussed in the second chapter. Then the learning from experience and the management of motion experience are talked about in the third chapter. Finally, how to use the learned motion experience to serve the verification of accessibility in an interaction loop is shown in the fourth chapter. The fifth chapter is dedicated to the conclusion.

- **Chapter 1: Problem Introduction and Literature Review** will firstly enumerate different kinds of accessibility verification problem in the product lifecycle. Then the mathematical description to the accessibility verification problem is demonstrated to promote the further discussion about its two available means of solution: the automatic algorithm and the manual

manipulation. The bottlenecks for the two means of solution are analyzed respectively. Beyond these, a novel perspective about integrating the motion learning to solve the accessibility verification is proposed. And the detailed comparison of the related motion learning methods is given.

- **Chapter 2: Interactive Motion Planning** will present and analyze the previous work which involves integrating the automatic motion planning algorithms with human manipulations. The retraction methods as an effective assistance to the sampling based planners will be explored and a novel random retraction method will be proposed. Then the convergence of the novel retraction method will be analyzed. Some limitations to the manipulation which guarantees the robustness of the retraction process will be talked about. Then a novel interactive motion planning method will be proposed in this chapter, which integrates humans' manipulation, the random retraction method and the sampling based algorithm. Due to the addition of the retraction method, the user's manipulation in the difficult region can be somewhat released from the burdensome operations by allowing some limited penetrations between the object and the obstacles. The final experimental results will show that the novel interactive motion planning method can efficiently reduce the time of accessibility verification in some difficult tasks.
- **Chapter 3: Motion Learning Based on Scenario Matching** will firstly introduce a hypothesis about the correlation between the motion path and the topological structure and the volume size of scenario. Since the topological representation of a scenario is a 1D curve, the similarity measurement of scenarios is converted to the curve matching problem. The scenario in this thesis is represented by the maximal sphere sequence which can be efficiently generated. The centerline and the radius change of the sphere sequence provide two features for the comparison of scenarios. Based on the two features, the similarity of scenario can be measured by the Dynamic Time Warping (DTW) method. For the reason of efficiency and robustness of scenario comparison, the scenario is split by the proposed segmentation method. The motion segments with their scenario segments will be gathered into the motion library according to their similarity. In the motion library, the proposed method will organize the motion segments with the R-tree structure. Due to the hierarchical and balanced structure of the R-tree, the real-time searching in a motion library with millions of motion segments is possible. Since some modules of the proposed motion learning framework contains many wide research fields, the focus in this chapter is on the proposition of the novel framework.
- **Chapter 4: Interactive Motion Planning with the Learned Experience** will present how to use the learned experience in the process of an interactive accessibility verification process. Based on the learned experience, the user's mission of manipulation can be converted to selecting the scenario. A modified PRM (Probabilistic Roadmaps) algorithm is used to guide

the exploration of users' manipulation. During the manipulation, only one point in the workspace is controlled by the user, in which manner the user will be released from the rotational operations of non-convex objects. After the scenario selection, a complete process will be started to divide the scenarios, and to retrieve, transform and repair the learned motion segments to fit each new scenario segment. A novel method which can efficiently determine the inaccessibility of the scenario according to the learned experience will also be proposed in this chapter. The rest of this chapter will discuss the possible extension of our motion learning and generation method to the articulated and differential model.

- **Chapter 5: Conclusion and Perspective** will briefly summarize all the methods and the contributions of this thesis, and provide a perspective for the future work.

# 1 Problem Introduction and Literature Review

---

1.1. Accessibility verification in the product design .....	5
1.1.1. Problem formulation .....	8
1.1.2. Configuration space .....	9
1.2. Accessibility verification by automatic algorithm.....	11
1.2.1. Sampling based motion planners.....	11
1.2.2. Narrow passage problem.....	12
1.2.3. Complete planning .....	13
1.2.3. Ergonomics analysis.....	13
1.3. Accessibility verification by manual manipulation .....	14
1.3.1. Manual manipulation in virtual environment with interactive devices .....	14
1.3.2. Constraints for humans .....	17
1.4. Introduction to the motion learning .....	18
1.4.1. Learning in the configuration space .....	20
1.4.2. Learning in the workspace .....	22

---

## 1.1. Accessibility verification in the product design

Globalization coupled with product customization and short-time to market has spearheaded new levels of competition among manufacturers, thus the verification and validation of engineering designs are of primary importance as they directly influence production performance and ultimately define product functionality and customer perception (Maropoulos and Ceglarek 2010). The efficiency of design process usually degrades dramatically with the increasing frequency of failure in the verification phase. An important factor considered in the verification phase is the accessibility.

What is Accessibility? Consider a mechanical assembly consisting of interacting parts. Is a given part can access a cluttered place without moving the other ones (Laumond 2006) ? The problem of accessibility is general among the whole product lifecycle. See Fig. 1.1 for a global view.

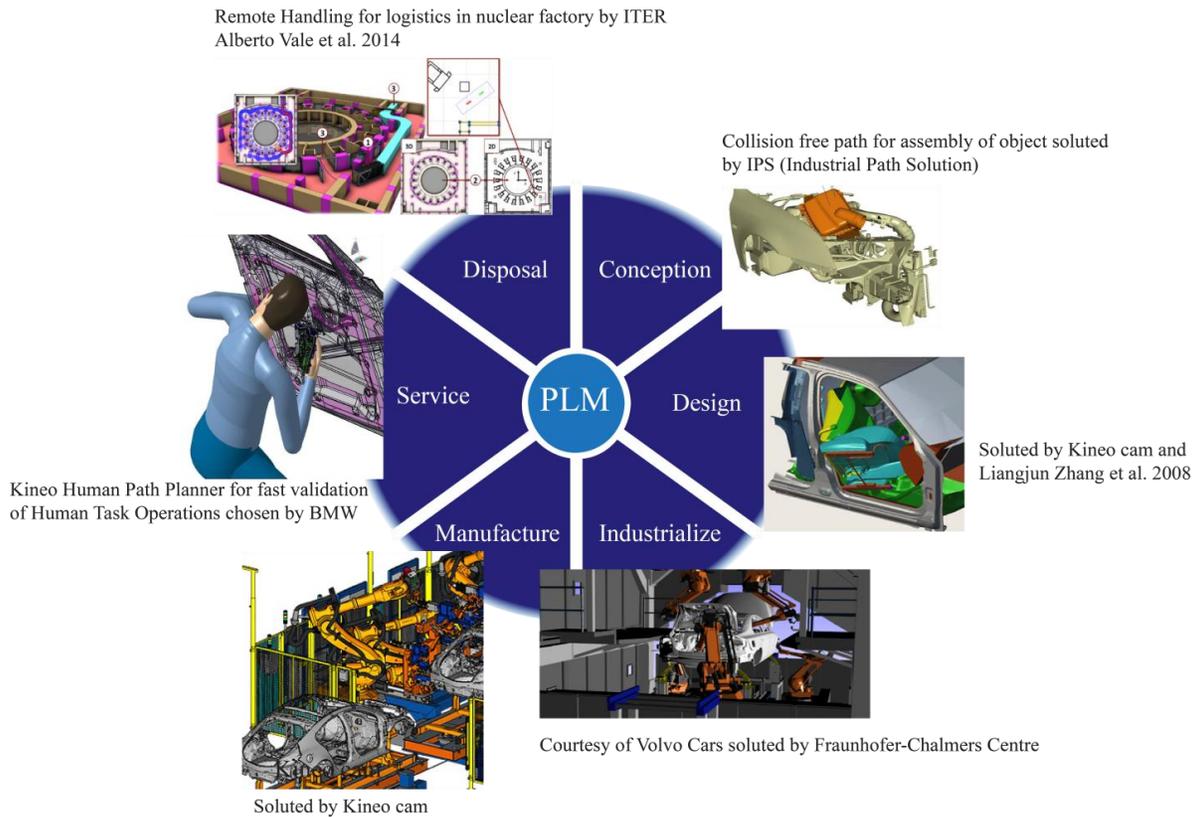


Figure 1.1. Accessibility verification in the product lifecycle

During the conceptual and preliminary design stages, the simulation of assembly and disassembly attempts to move a particular part without collisions between a free location outside the assembly and a predesigned assembly location. The predesigned assembly location of a component is the state at which it will be in contact with its mating component, namely the start of composing operation where these two components are going to be brought together by the selected joining method (Hsu and Lin 2002). The move involves computing the collision-free path for the rigid object with six degrees of freedom (DOF) among stationary obstacles (Garber and Lin 2002, Ferré Laumond et al. 2005, Zhang, Huang et al. 2008). Since the significant design changes are often very costly, hence the verification of accessibility can help designers know what the minimal structural changes are to make sure objects do not collide with each other (Shellshear, Tafuri et al. 2014).

The accessibility verification problems also arise in the layout design. In the layout design, the accessibility deals with the required vacant space locating around the component, and is used to guarantee the correct mechanism of the component. Moreover, the accessibility requirement means that each component has to be attainable from the container's entry (Bénabès, Bennis et al. 2010, Bénabès 2011). Recall the famous "piano movers' problem" as an example (Schwartz and Sharir 1983), the design of the locations of furniture and the entry of room can determine whether the piano can access the destination.

For the industrialization and manufacturing, the accessibility verification helps designers detect the colliding or invalid kinematic errors of implementing the tools and robots in an early conceptual design phase. (Michalos, Makris et al. 2010) investigated the accessibility constraints introduced by the use of robots in a cooperative assembly simulation. (Hermansson, Bohlin et al. 2013) focused on the accessibility of deformable parts in the virtual manufacturing, such as the collision-free mounting of connectors in a wire hardness installation in an already densely packed vehicle. Moreover, the accessibility analysis is necessary to determine how to orient the CMM (coordinate measuring machine) probe to approach certain features or points, and which probe may be used to access a certain point without collision (Spyridi and Requicha 1991, Spyridi 1994, Spitz, Spyridi et al. 1999, Spitz and Requicha 2000). In (Jang, Moradi et al. 2008), the accessibility analysis is used to determine the set of all collision-free paths of the gripper from the access poses to the grasp poses in a cluttered environment.

In service and disposal stage, the accessibility verification is always considered in the maintenance operation in industrial facilities. (Chang and Li 1995) firstly proposed using the automated motion planning method to verify the accessibility in the disassembly maintainability study. The problem also arises when workers are forced to adopt ergonomically stressful, unreachable or kinematic-broken postures in connecting, disassembling, filling, stalling or repairing activities. In these cases, while the collision-free trajectory for the component may be feasible, the operation cannot be performed because of the interference between the arm or body of a worker with the obstacles, the shortage of the sufficient space for the arm, or the shortage of the proper visibility of the work area (Rajan, Sivasubramanian et al. 1999, Geng, Zhou et al. 2013). (Park, Choi et al. 2011) dealt with the accessibility verification of a remote manipulator for a maintenance task in nuclear facilities.

In the context of logistics, the accessibility verification address the problems about whether a given body (e.g., tools, parts, loads...) can be moved from a place to another one by using the given transportation machines (e.g., manipulator robots, gantries, mobile platforms...) (Simón, Laumond et al. 2001). The challenge is to compute the collision-free motion in tight-fitting scenarios with the extremely cluttered environment and different combinations of nonholonomic, mechanical or kinematic constraints introduced on the transportation machines. For example, the (Lamiroux,

Laumond et al. 2005) dealt with the accessibility verification of the trailer-truck on the specialized road loaded with the very large size freight, in which the trailer trucks are subject to rolling-without-slipping and going-only-forward constraints. (Lamiriaux, Bonnafous et al. 2003) extended the complexity with considering two trailers. (Fonte 2011, Vale, Fonte et al. 2014) addressed the accessibility problem for an autonomous rhombic vehicle that operated in the buildings of the International Thermonuclear Experimental Reactor (ITER), in which the rhombic vehicle was modeled with two pairs (one for spare purposes) of drivable and steerable wheels positioned on the front and rear of the vehicle and two swivel wheels on the sides.

Computing a collision-free motion path for a particular object between different states (e.g., positions, orientations, poses...) is essential for all of the accessibility verification problems. The accessibility verification can be also regarded as the subset of the motion planning problems, however, the optimizations for energy and time are not always considered here.

### 1.1.1. Problem formulation

The accessibility verification involves computing a path in the object's *state space* without colliding with the obstacles. Usually the underlying objects (e.g., assembling part, worker's body, repairing tool, furniture...), noted as  $\mathcal{A}$ , and obstacles are complex and may be represented using thousands of polygons, and the models are given as polygon soup models with no connectivity or topology information. The 2D and 3D Euclidean space in which the objects move is called the *workspace*, which is noted as  $\mathcal{W}$ ,  $\mathcal{W} = \mathbb{R}^2$  or  $\mathcal{W} = \mathbb{R}^3$ . The region of the workspace which is occupied by the obstacles refers to the *obstacle region* noted as  $\mathcal{O} \subset \mathcal{W}$ . The *state space* describes all possible states that an object can have. The *state* could, for example, refer to the position and orientation of an assembling part or the pose of a human body. The *state space* is also called the *configuration space* noted as  $\mathcal{CS}$ , in which every state of the object is represented as a point in the configuration space. A configuration  $q \in \mathcal{CS}$  is free if the moving object placed at  $q$  does not collide with the obstacles or with itself. The *collision-free space*  $\mathcal{CS}_{free}$  is defined as the set of all free configurations in  $\mathcal{CS}$ .

$$\mathcal{CS}_{free} = \{q \in \mathcal{CS} \mid \mathcal{A}(q) \cap \mathcal{O} = \emptyset\} \quad (1.1)$$

Accordingly, the set of all configurations in which the object is in-collision with the obstacles refer to the *configuration space obstacle* noted as  $\mathcal{CS}_{obstacle}$ ,  $\mathcal{CS}_{obstacle} = \mathcal{CS} \setminus \mathcal{CS}_{free}$ . For the accessibility verification problem, the obstacles are usually stationary in the workspace. The problem also contains an initial configuration  $q_I \in \mathcal{CS}_{free}$  and goal configuration  $q_G \in \mathcal{CS}_{free}$ .

So the accessibility verification problem can be formulated as computing a continuous path:

$$\tau : [0, 1] \rightarrow \mathcal{CS}_{free}, \text{ such that } \tau(0) = q_I \text{ and } \tau(1) = q_G$$

or correctly reports that such a path does not exist. (LaValle 2006) indicated that finding a feasible solution for such kind of problem is challenging, which is PSPACE-hard.

### 1.1.2. Configuration space

Using the configuration space to describing the states of moving objects is proposed by (Lozano-Perez 1983), in which each coordinate of the configuration space represents a degree of freedom in the position or orientation of the object (Wenger and Chedmail 1991). In Fig. 1.2 (left), a triangulated object can only translate in 2D workspace. For this object, its configuration space  $\mathcal{CS}$  is 2D Cartesian space  $\mathbb{R}^2$  in which each configuration  $q \in \mathcal{CS}$  is defined by the  $x$  and  $y$  coordinates of the reference point on the object,  $q = (x, y)$ . The configuration space obstacle  $\mathcal{CS}_{obstacle}$  for this kind of object can be easily computed by Minkowski sum (Latombe 1991), shown as the region with diagonal texture in the Fig. 1.2 (right).

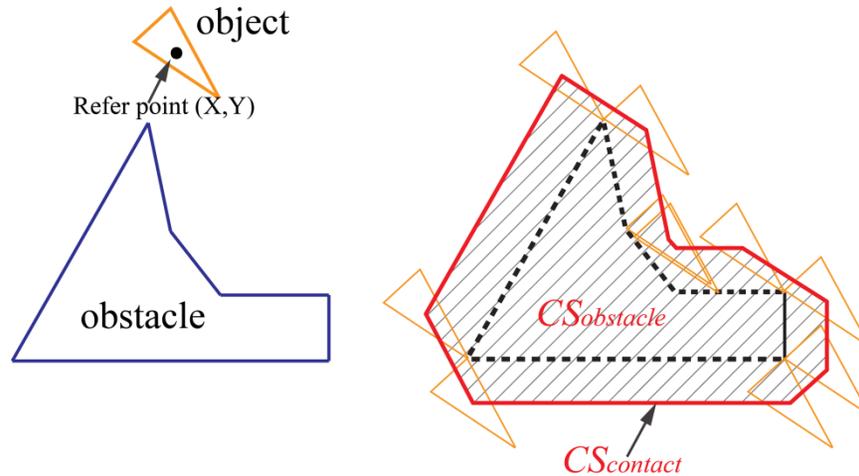


Figure 1.2. Configuration space of a 2D translating object

For a rigid object which is able to translate and rotate in 2D plane, its configuration space  $\mathcal{CS}$  is the product of  $\mathbb{R}^2$  and  $SO(2)$ ,  $\mathcal{CS} = \mathbb{R}^2 \times SO(2)$ . Such an example is shown in Fig. 1.3, in which the configuration  $q$  of the “L” shape object is represented as  $q = (x, y, \theta)$ . The *contact space*  $\mathcal{CS}_{contact}$  of this object is computed through checking the collision situations of the uniformly sampled configurations, and is shown in Fig. 1.3 (right). The *contact space* is defined as the set of configurations at which the object barely touches one or more obstacles without penetration, and it can also be regarded as the boundary of the configuration space obstacle  $\mathcal{CS}_{obstacle}$ .

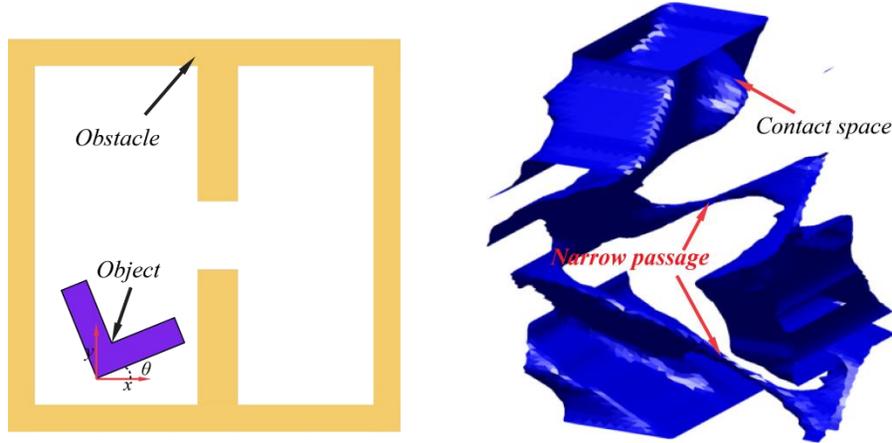


Figure 1.3. Configuration space of an object freely moving in 2D plane

In 3D workspace, if a rigid object can translate and rotate, its action usually contains six degrees of freedom. As shown in Fig. 1.4, three coordinates  $x, y, z$  in 3D Cartesian space represent the position of this object, and another three parameters  $\alpha, \beta, \gamma$  are usually used to represent its orientation, where  $\alpha, \beta, \gamma$  are the quantities of counterclockwise rotation angle about the  $z$ -axis,  $y$ -axis and  $x$ -axis respectively. Thus, the configuration space  $\mathcal{CS}$  of a free-flying rigid object is 6-dimensional and equals to the Cartesian product of two manifold subgroups: the group of translations  $\mathbb{R}^3$  and the group of rotations  $SO(3)$ ,  $\mathcal{CS} = \mathbb{R}^3 \times SO(3) = SE(3)$ .

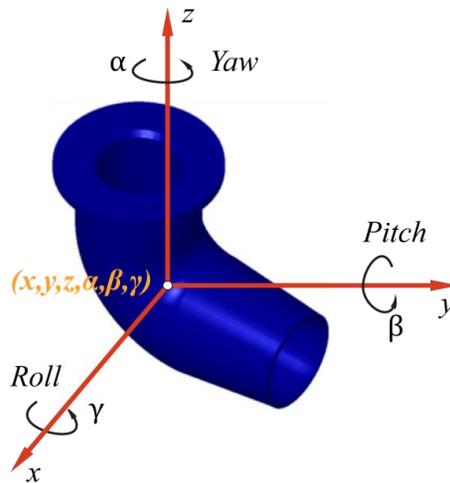


Figure 1.4. Configuration of a free-flying rigid object in 3D

The precisely computation of collision-free space  $\mathcal{CS}_{free}$  is very challenging. A general approach for configuration space computation proceeds by enumerating contact surfaces for every pair of features from the object and the obstacle (Varadhan, Kim et al. 2006). Each contact surface is the locus of configurations of an object at which a specific boundary feature of the object (i.e. vertex, face, or edge of a rigid object represented as polyhedra) is in contact with a boundary feature of the

obstacles. The combinatorial complexity of the entire  $\mathcal{CS}_{free}$  can be  $O(n^k)$ , where  $n$  is the number of contact surfaces and  $k$  is the dimension of the configuration space (Zhang 2009). For a non-convex rigid object which is a polyhedron with  $p$  polygons and is able to translate and rotate in a polygonal environment bounded by  $m$  edges, the combinatorial complexity of  $\mathcal{CS}_{free}$  is  $O((pm)^6)$ . The recent breakthrough of computing in reasonable time the 6-dimensional configuration space of a rigid object is realized by (Nelaturi 2011, Nelaturi, Lysenko et al. 2012), in which the Fast Fourier Transformation (FFT) is used to efficiently compute the convolution of Minkowski sums with rotations.

## 1.2. Accessibility verification by automatic algorithm

On account of the complexity, the motion planning algorithms usually avoid the explicit construction of  $\mathcal{CS}_{obstacle}$  through sampling in the configuration space. During the late nineteen nineties, sampling based strategies were proposed as a breakthrough to provide practical solutions to the motion planning problems.

### 1.2.1. Sampling based motion planners

The sampling based motion planners take samples in the object's configuration space and connect these samples by a graph which captures the connectivity of the  $\mathcal{CS}_{free}$ . This kind of algorithms usually consists of three components: the sampling strategy, the local connection and the path query. These methods are said to be *probabilistic complete*, which means that the probability of finding a solution, if one exists, converges to 1 as the computing time tends to infinity (Jaramillo 2007). Depending on the strategy of constructing the graph, the sampling based motion planners are categorized into two families: Probabilistic Roadmaps (PRM) (Kavraki, Svestka et al. 1996) planners and Rapidly Exploring Random Trees (RRT) (Lavalle 1998) planners.

The main idea behind the PRM is to firstly take random samples in the object's configuration space. Then the collision detection is implemented to test whether the sampled configuration is in the free space  $\mathcal{CS}_{free}$ . If the sampled configuration lies in  $\mathcal{CS}_{free}$ , it is connected by local planner to some neighbors, typically either the  $k$  nearest neighbors or all neighbors less than some predetermined distance. Configurations and connections are added to the graph until the roadmap covers the connectedness of the space. In the query phase, the start and goal configurations are connected to the graph, and the path is obtained by a Dijkstra shortest path query. Since its creation, the details of basic PRM have been studied and extended by many variants, a comprehensive comparison between them can refer to (Geraerts and Overmars 2004). The PRM framework is suitable to solve

multiple-query problems, namely providing multiple initial configurations or goal configurations with the same object and obstacles.

In contrast, RRT planners are more appropriate for single-query problem. RRT is designed to efficiently search configuration space by randomly building an open-loop graph, or more precisely a space-filling tree. The tree is usually rooted at the initial configuration and is incrementally constructed by attempting the connection between the randomly sampled configuration and the nearest configuration in the tree. If the connection completely lies in  $\mathcal{CS}_{free}$ , the sample and the connection are added to the tree; otherwise, a valid part of the connection extended from the nearest configuration is added. An important property of RRT is that it is constructed by iteratively breaking large Voronoi regions into smaller ones, which results in the expansion is heavily biased toward unexplored portions of the configuration space (Lavalle, Kuffner et al. 2000).

The sampling based planners has been successfully used to solve many practical problems with high degrees of freedom (Latombe 1999). However, they all have some level of difficulty if the object's free space  $\mathcal{CS}_{free}$  has narrow passages (Denny and M. Amato 2013).

### 1.2.2. Narrow passage problem

Narrow passages are small regions whose removal changes the connectivity of the configuration space (Hsu, Kavraki et al. 1998, Sun, Hsu et al. 2005). The narrow passage problems usually arise when complex assemblies are composed of large number of parts, which often lead to scenes with high obstacle densities (Morato, Kaipa et al. 2013). The narrow passages occupy small volume in  $\mathcal{CS}_{free}$ , thus any sampling distribution based on the volumes is likely to fail (Hsu, Jiang et al. 2003). An example of narrow passages is shown in Fig. 1.3 (right), in which two narrow passages correspond to the “L” shape part passing the hole with clockwise and counterclockwise rotations. Any found path is essential to translate through one of the two passages. However, these two passages occupy so small volume which makes the search for a complete path extremely difficult.

Different heuristics have been created to bias the sampling based planners to overcome narrow passage problem. *Decomposition based planners* decompose the configuration space into different kinds of cells and get an approximate corridor with A\* algorithm, then explore in this corridor to find a collision-free path (Rosell, Vázquez et al. 2008, Ladeveze, Fourquet et al. 2009). *Obstacle based planners* either plan motion compliant to C-contact space with geometric formulations (Ji and Xiao 2001, Peng Tang and Jing Xiao 2008), or increase sampling around the obstacle (Amato, Bayazit et al. 1998, Boor, Overmars et al. 1999, Zheng, Hsu et al. 2005, Hsin-Yi, Thomas et al. 2012) and on the C-contact space (Redon and Lin 2005). In (Rodriguez, Xinyu et al. 2006), vectors generated on triangulated meshes of CAD model are used to guide the extension of basic RRT planner. *Retraction based planners* attempt to improve the sampling in narrow passages through

retracting in-collision configuration to the nearest configuration on C-contact space (Saha, Latombe et al. 2005, Liangjun and Manocha 2008) or the medial axis of free space (Wilmarth, Amato et al. 1999, Wilmarth, Amato et al. 1999, Lien, Thomas et al. 2003). Despite some improvements, narrow passages are still the major issue for sampling based planners. In fact, adding the consciousness of the local changes of workspace to algorithms will endure some expensive computation.

### 1.2.3. Complete planning

The complete planning issue arises when sampling based motion planners are applied to solve the accessibility verification problem. Compared to finding an exact motion, accessibility verification concerns more on the feasibility. An algorithm is considered complete if for any input it correctly reports whether there is a solution in a finite amount of time. However, such completeness is not achieved with sampling-based planning algorithms (LaValle 2006). If no collision-free path exists, sampling based algorithms may run for a long time before certificating that (Zhang, Kim et al. 2008). Due to the combinatorial complexity of the configuration space, some algorithms attempted to solve complete motion planning problem through the adaptive cell decomposition of the configuration space (Varadhan, Krishnan et al. 2005, Varadhan and Manocha 2005, Zhang, Kim et al. 2007, Zhang, Kim et al. 2008, Zhang, Kim et al. 2008). The subdivision of cells relies on efficient cell labeling algorithms which classify the cells into the empty cell (completely contained in  $\mathcal{CS}_{free}$ ), the full cell (completely contained in  $\mathcal{CS}_{obstacle}$ ), and the mixed cell. Then a graph is constructed by connecting the empty cells and mixed cells. The non-existence of a path on the graph provides a sufficient condition for path non-existence between the initial and goal configuration in  $\mathcal{CS}_{free}$ . Although these methods contribute motivating solutions to the complete motion planning, they cannot prevent unnecessary subdivision of cells and the complexity is increased of the non-convex object or degrees of freedom. As a result, the methods are limited to the rigid object with 3 or 4 DOFs.

### 1.2.3. Ergonomics analysis

For the accessibility verification, another challenge for designers is to test and verify the geometric constraints to the articulate system of human skeleton. A direct method is adding a digital human mannequin into the reasoning loop of motion planning algorithm. Combined with inverse kinematics, (Ferré Laumond et al. 2005) successfully planned one hand manipulation with 7 DOFs and two hands manipulation with 15 DOFs of a virtual mannequin in less than one minute. However, in considering whole body delicate motion planning, up to 32 degrees of freedom will load a huge amount of calculation. Besides, the quality of generated motion of mannequin becomes a major issue, which mainly concerns reducing the redundant degrees of freedom and making the motion look more natural.

In addition, a key challenge for the designers is to analyze end-users' response in the design process and promote product innovation, but users' preferences are generally unable to precisely formulate (Poirson, Petiot et al. 2013). Thus, optimization for this issue with the automatic method is extremely difficult at present, and integration with the direct participation of designers is an essential stage (Bénabès, Poirson et al. 2013, Lu and Petiot 2014).

### **1.3. Accessibility verification by manual manipulation**

The participation of designers in the accessibility verification process is in forms of manual manipulations of digital prototype in the virtual reality environment. Virtual Reality (VR) is a scientific and technical domain exploiting the possibilities of computers and behavioral interfaces to simulate in a virtual world the behavior of 3D entities, which interact in real time with each other and one or more users in pseudo-natural immersion through sensorimotor channels (Fillatreau, Fourquet et al. 2013). Virtual prototyping (VP) is the application of virtual reality for prototyping physical mock-ups (PMUs) using product and process data, of which the idea is to replace, at least partly, physical mock-ups (PMUs) by software prototypes (Gomes de Sá and Zachmann 1999). The virtual reality tools are intensively studied and applied to the design verification and validations in the product lifecycle (Jayaram, Jayaram et al. 2007, Maropoulos and Ceglarek 2010, Keller, Doceul et al. 2013).

Since the manual manipulation of digital prototype is usually realized by capturing humans' motion with the interactive devices. The discussion of this section is related to the limitations of manual manipulations in the accessibility verification process, which arises with the limitations of the available interactive devices and the constraints of humans' motion planning capabilities.

#### **1.3.1. Manual manipulation in virtual environment with interactive devices**

In accessibility verification tasks, the human motion is captured and transferred by the interactive devices to the control parameters which continuously change the position and orientation of the CAD model in the digital world. Depending on the task, the interactive devices are required to provide different number of degrees of freedom (DOF). If the task only involves the motion in 2D plain (e.g. pushing furniture, steering models of vehicles or aircrafts), two DOFs are required to represent the position of object and one additional DOF is needed for rotation. The 2D mouse and joystick which provide two or three simultaneous DOFs can fulfill most of these requirements. For manipulating rigid objects in 3D, at least six DOFs are required, in which three DOFs are used for the position and another three DOFs are used for the orientation. Based on whether force reaction

signals can be send back to users, the interactive devices for controlling 3D object can be divided into two categories: the non-haptic device and the haptic device.

For non-haptic devices, the result of manipulation is usually only demonstrated visually. The commercial devices includes 3Dconnexion's 3D mouse (<http://www.3dconnexion.com/>), ART's Flystick (<http://www.ar-tracking.com/>), Cyber Glove Systems' data-glove (<http://www.cyberglovesystems.com/>) and motion capture systems. The 3D mouse is non-portable which is only suitable for the desktop application, and the others are portable and extendible for applications in the larger workspace. The 3D mouse and Flystick can only capture movements of six DOFs, whereas data-glove and motion capture systems can capture more complex movements of hand's fingers and human body's joints. Based on whether the marker is used for capturing the motion, the motion capture systems can be divided into: the marker based system and the marker-less system.

For the marker based motion capture system, the combinations of special markers are usually attached to the object to represent the position and orientation of the object. Some of the markers either reflect (ART <http://www.ar-tracking.com/>, Vicon <http://www.vicon.com/>, OptiTrack <https://www.naturalpoint.com/optitrack/> and IoTracker <http://www.iotracker.com/>) or emit IR (infrared ray) signals to the measuring cameras. The others use pulsed DC magnetic sensors (Ascension Technology <http://www.ascension-tech.com/>) or the group of gyroscopes and accelerometers (Xsens <http://www.xsens.com/>) as markers. The marker based motion capture system can offer good measurement accuracy and update rate, but it has drawbacks including: (i) it is time-consuming to put markers onto the tracked object; (ii) the attached markers may interfere with the object's normal movement; (iii) and the high acquisition cost (Leu, ElMaraghy et al. 2013).

The marker-less motion capture system either use stereo RGB camera, silhouette technique, the structured light technique or depth sensor to capture, estimate and recognize the pose of human body (Organic Motion <http://www.organicmotion.com/>, Kinect <http://www.microsoft.com/>, Intel® RealSense <http://www.intel.com/>, and SoftKinetic <http://www.softkinetic.com/>). Compared to the marker based system, the marker-less based system has lower cost and more natural interface, but has lower accuracy and robustness.

The non-haptic devices provide natural interface and can capture very complex movements of human body, which makes them suitable for some ergonomics studies in the loosely constrained environment. However, for the accessibility verification task, the movements of the non-convex object are always carried out in the cluttered scenes, which makes the simulation poorly visible or the virtual object partly or totally occluded. (Leu, ElMaraghy et al. 2013) indicated that the haptic feedback becomes essential for these situations. Some research has also concluded that the addition

of force feedback to the virtual environment allowed the participants to complete the task faster (Popescu, Burdea et al. 1999, Volkov and Vance 2001, Seth, Vance et al. 2011, V az, Lozano-Rodero et al. 2013).

Haptic devices are the advanced technology that redoubles the simulation effects, which approximates real-world sensations in virtual, tele-operable, and hazardous environments (Park, Choi et al. 2011). Haptic feedback can be categorized into force feedback and tactile feedback. Force feedback relates to a virtual object’s hardness, weight and inertia, while tactile feedback simulates the user’s feel of the virtual object’s surface geometry, smoothness, slippage, temperature (Burdea 1999). Force cues provided by the haptic technology are created when an object crushes or penetrates into the obstacles, and can help designers feel and better understand the virtual objects by supplementing visual and auditory cues and creating an improved sense of presence in the virtual environment (Seth, Vance et al. 2011). The professional desktop haptic devices usually include Sensable Technologies’ PHANTOM from Geomagic (<http://geomagic.com/>), Haption’s Virtuouse (<http://www.haption.com/>), and Novint’s Falcon (<http://www.novint.com/>). These devices can provide not only force feedback in three translational degrees of freedom but also torque feedback in three rotational degrees of freedom. But, these devices also have some drawbacks like: (i) the workspace limitations result in the restricted user motion in the environment; (ii) these devices need to be stably mounted, thus the user is constantly attached to a handle, their use with the bimanual manipulation and the associated whole body movement in immersive virtual environment becomes unfeasible (Seth, Vance et al. 2011, V az, Lozano-Rodero et al. 2013).



Figure 1.5. Some models of haptic devices

The wearable haptic devices such as CyberTouch, CyberGrasp (<http://www.cyberglovesystems.com/>) or Rutgers Master II (Bouzit, Burdea et al. 2002) can exert grasp forces that are roughly perpendicular to the fingertips. With these devices, users are able to feel the size and shape of the CAD models. Since only the fingers are constrained in collision and whole hand is freely movable, these devices cannot prevent the grasped object from penetrating or crushing other virtual objects, and thus are suitable for tasks that only involve dexterous manipulations. CyberForce (<http://www.cyberglovesystems.com/>) can be regarded as the

combination of CyberGrasp and desktop haptic device. It provides a very natural haptic interface for hand manipulations, with which the users are able to sense the object's shape and size as well as its mass and inertia. Because all haptic feedback devices incur significant costs and have strict geometry, placement and workspace requirements, these prevent them from being used widely (Leu, ElMaraghy et al. 2013). The comprehensive comparison of interactive devices can refer to Tab. 1.1. In the table, the "Simultaneous DOFs" refers to the number of degrees of freedom that the device can provide in once operation. The "Portable" refers to that the device is not stably mounted or connected by a cable with very limited action range. As we can see, no device can fulfill all the requirements. The choice of interactive devices depends on the specific manipulation task.

Table 1.1. The comparison between different interactive devices

Device		Total DOFs	Simultaneous DOFs	Accuracy	Tactile feedback	Force feedback	Portable	Price
2D mouse		3	2	High	N	N	N	Low
Joystick		4	3	High	N	N/Y	N	Low
Flystick		6	6	High	N	N	Y	Low
3D mouse		6	6	High	N	N	N	Low
Motion capture system	Passive marker	>6	>6	High	N	N	Y	High
	Active marker	>6	>6	Medium	N	N	Y	High
	Marker-less	>6	>6	Low	N	N	Y	Low
	Electromagnetic	>6	>6	High	N	N	Y	High
	Inertial	>6	>6	High	N	N	Y	High
Desktop haptic device		3~6	3~6	High	N	Y	N	Medium
Data glove		>6	>6	High	N	N	Y	High
CyberTouch, CyberGrasp, Rutgers Master II		>6	>6	High	Y	N	Y	High
CyberForce		>6	>6	High	Y	Y	N	High

### 1.3.2. Constraints for humans

Although the virtual reality technique and interactive device provide the possibilities to evaluate issues related to the ergonomics, the manual manipulation is time-consuming for some tasks with small tolerance between the manipulated object and the obstacles, or poor visibility. For example, (Ferré Laumond et al. 2005) has shown that, for the task of disassembling a wiper motor from the car body (Fig. 1.6 (a)), the automatic algorithm found the solution within one minute whereas a manual search took around 2 to 5 days of an experimented operator. Since the manipulated object is highly non-convex in this kind of tasks and the scene is only partially visible to the humans, it is difficult for humans to solve the problem even in real life. Another similar example is the famous "hedgehog puzzle" which is shown in Fig. 1.6 (a) right. The puzzle consists of a small sphere with

protruding spikes of various lengths contained within a cylinder perforated with holes of different sizes. The challenge posed by the puzzle is how to release the sphere (the hedgehog) from the cylinder (the cage). This puzzle usually spends several hours or days to solve for the players without any prior knowledge.

However, humans have some advantages in planning the motion globally. It is easier for humans to know “where to go” than “how to go”, whereas the situation is opposite for the automatic algorithms. For example in Fig. 1.6 (b) left, the mission involves inserting a stick into a bug-trap. The solution to this mission is obvious to humans but difficult to the automatic algorithm. If the bug-trap is designed more complex like in Fig. 1.6 (b) right, the difficulty of finding the path between initial and goal configurations has not much increase for humans, but it is extremely difficult for automatic algorithms. The available solutions, which attempt to enhance both humans and automatic algorithms in a motion planning task, allow the user and the algorithm work in a complementary fashion. This topic will be deeply discussed in the second chapter. In the next section, we raise another discussion which can lead to an efficient solution to the accessibility verification in a new perspective.

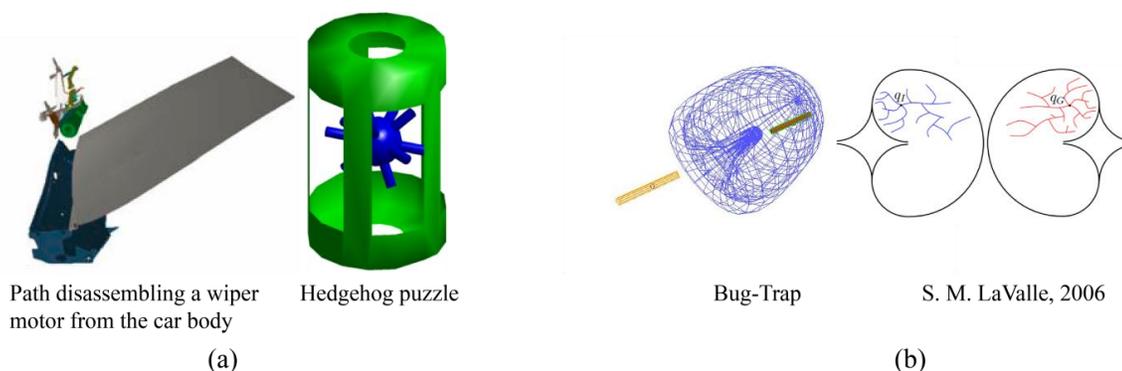


Figure 1.6. Advantage and disadvantage of human's capability of motion planning

## 1.4. Introduction to the motion learning

Although humans do not possess the capabilities of accurately computing the complex movements in some difficult situations, they show some talents to make very complex movements in reaction to the change of the environment, and this reaction is usually made in a very short time. For example, Fig. 1.7 shows the captured frames of a dexterous shooting action of a famous football player. The action from the video game is produced by firstly recording the movements of a real football player with the motion capture system introduced in the Section 1.3.1. The movement is so complex that it involves the appropriate running velocity, good body's balance keeping, perfect cooperation of four limbs and accurate choice of the contact point between the foot and the ball. More than thirty

degrees of freedom are needed for representing a movement like this. It could be imagined how a huge burden of computation should be loaded onto an algorithm when computing the movement. However, the real football player makes this movement instantly according to the situation in the match. An important factor behind this distinction are that humans can efficiently learn from experience, recognize similar situations and transform the memory motion according to the learned skill. And the mapping from the memory experience to the new situation can be made naturally and robustly.



Figure 1.7. The captured frames of a dexterous shooting action of a football player (Courtesy of <http://axeetech.com/>)

It also makes sense to learn motion experience in the theme of manipulation. Recall the “hedgehog puzzle” introduced in the last section, one can get the edge on the others in solving the puzzle if he has solved it before. Generally, if someone is constantly trained with the similar object and manipulation environment, he can finish the task faster and faster. However, the implicit expertise or knowledge behind the manipulation is difficult to express and formulate. The research from neurobiology has led many to speculate that the information-processing abilities of biological neural systems must follow from highly parallel processes operating on representations that are distributed over many neurons (Mitchell 1997). The simulation of this parallel and distributed operating mechanism is still far from realization at present. Instead, the research work in the motion planning filed intends to explore the information in the previous planning process to reduce the computation or directly predict a motion path in the new task. In our work, we will present that the interactive motion planning method integrated with abilities of learning from experience can enhance the cooperation between users and automatic algorithms to simultaneously overcome their own drawbacks and strengthen their advantages, and thus more efficiently verify the accessibility in a product design process.

A lot of information can be accumulated in the accessibility verification tasks no matter the tasks have been solved by users or automatic algorithms. The information can include the found motion

path of object, the sampled configurations from the configuration space or the description of CAD model and their relative distance. The challenges of motion learning are how to develop a concrete parametric representation of the manipulation tasks, to evaluate the similarity of the task based on the representation and to train a function or policy to predict the motion path or a direct determination of the accessibility in new task. Furthermore, in contrast to some other application fields of motion learning (e.g. animation production), the learning strategies have to primarily consider the avoidance of collision with densely located non-convex obstacles. Depending on whether using the information in the configuration space or in the workspace to train the predictive function or policy, the motion learning methods can be divided into two categories. In the next two subsections, we will introduce the prior work in these two categories and present some differences between them and our method.

#### **1.4.1. Learning in the configuration space**

Although there are many motion planning techniques, there is no method that outperforms the others for all problem instances. Thus some methods use learned features in the configuration space to determine the sampling strategy. (Morales, Tapia et al. 2005) proposed a framework of the feature sensitive motion planning. The method recursively characterized and partitioned the configuration space into sub-regions according to the local features (e.g. free node ratio, components, edge ratio, node distance, edge length...). The regions were classified as homogeneous (free, cluttered, or narrow passage) or non-homogeneous by a trained decision tree. Then the roadmaps in the sub-regions were created by the best suited motion planners. Finally, the roadmaps in all the sub-regions were connected to capture the connectivity of the collision-free space. It has been demonstrated that their approach could outperform any of the individual planners on a variety of problem instances. (Morales, Tapia et al. 2005) improved this method by reducing the redundant region division and time spent on integrating the regional roadmaps. Similarly, some other algorithms also used the trained policies to decide the adaptive sampling strategy in the configuration space, such as Hybrid PRM planner (Hsu, Sánchez-Ante et al. 2005) and its variant (Lien 2008), and the approaches based on information theory (e.g., information gain and entropy) (Burns and Brock 2005, Burns and Brock 2005). The idea of adaptive sampling also appeared in some non-learning approaches, such as (Rodriguez, Thomas et al. 2008). While significant improvement has been shown over non-adaptive approaches, (Tapia, Thomas et al. 2009) indicated that these methods had all been seen to have serious drawbacks that limit their usefulness such as requiring significant user intervention (e.g., manual classifications of training instances for supervised machine learning methods, parameter tuning to set learning rates and learning weights) or restricting the types of problems they are able to solve. Moreover, although these methods adaptively apply motion planners or sampling strategies to the characterized regions of problem,

none improvement has been made to any single planner. And given similar environments, these methods still build the roadmaps from scratch (Lien and Lu 2009).

(Pan, Chitta et al. 2013) used instance-based learning to accelerate the collision-query of sampling based planners in avoidance of exact collision checking. In their method, all the results of prior collision queries, including collision-free as well as in-collision queries, were stored in a library. Then such prior information was sufficiently exploited to accelerate the planner computation by k-NN (k-nearest neighbor) queries based on the locality-sensitive hashing. This approach can be obviously superior to any sampling algorithms with the increased number of planning times in the same static environment. However, no similarity of different environments was assumed in their method.

In recent years, some work transformed the conventional strategies of sampling in the configuration space to the sampling in the path space by constructing a library of paths. (Branicky, Knepper et al. 2008, Knepper and Mason 2009) employed path sampling as part of a hierarchical planning strategy. The compact subset of a pre-computed path database can be reused to generate the optimized local path in avoidance of the collision with obstacles. Their method has been demonstrated robust and efficient for the models with paths comprising fixed-curvature arcs, e.g. autonomous mobile robots whose sensors can only scan a limited range (Howard, Pivtoraiko et al. 2014).

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 complex motions of virtual mannequins and humanoid robots. The motion primitive established on experience is an imitation of motion learning mechanism of human. These motion primitives can be represented as optimized stances (Kallmann, Bargmann et al. 2004, Hauser, Bretl et al. 2008), or differential equations called Dynamic Movement Primitives (DMPs) (Pastor, Hoffmann et al. 2009). The adoption of motion primitives aims to reduce the redundant degrees of freedom and decrease the search space.

The above works using path database for local motion planning have concentrated on constructing an offline and optimal static path sets, the challenge to these methods is to adaptively decide which path should be used.

(Berenson, Abbeel et al. 2012) proposed a motion planning framework that could learn from experience. In their method, the pre-collected paths can be efficiently retrieved to serve the planning in the whole task rather than in the local task. The dynamic database management based on the path similarity was also proposed. The idea of retrieving-repairing strategy is similar to ours. However, their path retrieval method is based on two criteria: (i) the distance between the endpoints of the path and the query start and goal; (ii) the amount of constraints violated by a path. No structural

similarity of task environment has been dealt with in their path retrieval strategy, which limits the adaptation of their learning method.

The development of motion learning based on the features of configuration space can be greatly impeded by the bottleneck that any operation in the configuration space will endure some expensive computation. Consequently, most of these methods are limited to exploring the information that comes from the collision-checking of sampled configurations, and none of the similarity of the shape and arrangement of obstacles has been fully considered. In the discussion of next subsection, another category of motion learning methods attempts to explore the relationship between the features of workspace and the motion path.

#### **1.4.2. Learning in the workspace**

The motion learning based on the features of workspace addresses the problem of whether it is possible to learn a function that maps directly from parameters describing an arrangement of obstacles in the workspace, to some information about the configuration space that can be used effectively by a sample-based planner (Finney, Kaelbling et al. 2007).

The similar concept has also been proposed in the community of imitation learning or learning from demonstration (LfD). Within LfD, a policy is learned from examples, or demonstrations, provided by a teacher, in which the examples are defined as sequences of state-action pairs that are recorded during the teacher's demonstration of the desired robot behavior, and LfD algorithms utilize this dataset of examples to derive a policy that reproduces the demonstrated behavior (Argall, Chernova et al. 2009). However, the focus is usually to extract motions from human demonstration of different tasks which can be later repeated “exactly” by robots, the generalization to different environments and collision avoidance with obstacles is rarely considered in the imitation process (Jetchev 2012). In contrast, the collision avoidance is usually considered as the most important factor in the accessibility verification tasks in which the cluttered scenes are very common.

Since the difficult regions in the configuration space often result from the constrained region in the workspace. Some approaches trained the predictive policies based on some parametric representation to the workspace. Then the trained policies can map a partial or global path to the new environment. (Finney, Kaelbling et al. 2007) trained a mixture of regression functions based on the collection of pairs of the instance description vector and partial path in the constrained region of workspace. The main drawback of this method comes from their parametric representation of the instance, for example, it is able to describe the same corridor in different positions and orientations but not the corridor with similar shapes. Therefore, it is not easy to estimate how much computation burden will be loaded onto the offline training process of the model when it is extended with the much more complex descriptor. (Jetchev and Toussaint 2013) provided another perspective which

predicted the whole trajectories rather than the partial paths from an offline constructed and optimized motion library. In their method, the task features have been described as the vectors which consist of a voxel representation of a scene and the relative distances between every part of the robot body and obstacles. When predicting, an index into the motion database has been proposed first to suggest an initial trajectory which is transferred and optimized in the latter phase to generate a whole trajectory between the initial configuration and the goal configuration. Although they demonstrated application instances in which their method can predict an optimal trajectory faster than the traditional sampling based method, there are still some drawbacks, such as their path library is constructed beforehand and it is static, which assumes that it is possible to envision the range of scenarios the robot encounters a priori (Berenson, Abbeel et al. 2012). In their paper, they also concluded that the design of the scenario has big effect on performance when the number of obstacles is more than 20.

Rather than predicting trajectories directly, (Dragan, Gordon et al. 2011) predict qualitative attributes (i.e. the goal configuration of the end-effector) of the trajectories first, and then map these qualitative attributes into initial guesses for a local optimizer which is developed in (Ratliff, Zucker et al. 2009). Since only the goal configuration has been considered, the success of collision avoidance depends uniquely on the optimizer. In accessibility verification tasks, the intensely constrained scenes usually generate some local minimal regions. It is not clear if the suggested initial path and the optimized results can efficiently avoid the collision in these tasks.

The framework of the above introduced learning-in-workspace methods can be concluded as training a predictive policy based on the parametric representation of the workspace. The key challenges for these methods are the design of appropriate and robust parametric representation, and the selection of efficient training strategies.

(Lien and Lu 2009) provided another perspective to record the roadmaps which capture the configurations of the object moving around the obstacles. Then these roadmaps can be reused by recognizing the similar obstacles. Thus, no matter how the arrangement of the obstacles is, if the obstacle can be recognized, the previous computed roadmap can be transformed and connected to guide the new motion query. One of the most critical bottlenecks of this method is the efficiency of the shape matching method which is based on the approximate convex decomposition of the shape. Moreover, the perturbation on the surface of obstacles may dramatically change the structure of configuration space.

In this thesis, we attempt to propose a novel motion learning paradigm. In contrast to the previous concerns on the arrangement of obstacles, our methods explore the relationship between the scenario and the configuration space, while the scenario is defined as the subset of the workspace

excluding the obstacles. In the difficult region of workspace, no matter how the arrangement of obstacles is, the motion of object is only decided by the scenario, thus the concept of scenario is more robust to describe the environment to the motion of objects. On further exploration, we propose a hypothesis that objects can perform similar motions in two scenarios which have similar topological information. Although an exact mathematical proof to the hypothesis is still in debt, a lot of empirical evidences are provided in this thesis to promote the further discussion in the scientific community. Our motion learning paradigm is established on the hypothesis and it has shown great improvements on the planning instances of free-flying rigid bodies. For the motion planning problems of object with articulated components, we predict that the appropriate affine transformations in configuration space are necessary.

# 2 Interactive Motion Planning

---

2.1. Introduction .....	25
2.2. Previous work in the interactive motion planning (IMP) .....	27
2.3. Retraction based on the penetration depth .....	29
2.3.1. Retraction based motion planner .....	30
2.3.2. Random retraction method .....	32
2.3.3. Estimation of the penetration .....	34
2.3.4. Convergence .....	37
2.3.5. Retraction results in applications .....	39
2.3.6. Conclusion .....	46
2.4. IMP based on the retraction and the local connection .....	46
2.4.1. System global view .....	47
2.4.2. Seeds connection by the RRT-connect algorithm .....	49
2.4.3. Application cases .....	50
2.5. Conclusion .....	57

---

## 2.1. Introduction

As discussed in the first chapter, humans and the automatic algorithms possess different advantages in planning the motion. Humans have the advantages in scenario recognition and global planning, but lose their efficiency in the cluttered environment which requires complex geometrical manipulations. In contrast, the algorithms can quickly find a local solution in the constrained environment, but have not enough intelligence in the global planning even being aided by various

heuristic methods. Moreover, user's preference and affection are more and more considered in the product design process, some of these subjects are still difficult to precisely model at present, which makes users' participations essential in the product life cycle.

Therefore, the cooperation of humans and motion planning algorithms provides a new perspective to the accessibility verification in the product design, which is not a simple addition of humans and algorithms, but a process integrating them together and empowering each other. The objective of integrating humans with automatic motion planners include:

- i. User manipulates digital models with guidance provided by automatic motion planning algorithm
- ii. The searching process of automatic motion planning algorithm is accelerated by human's manipulation

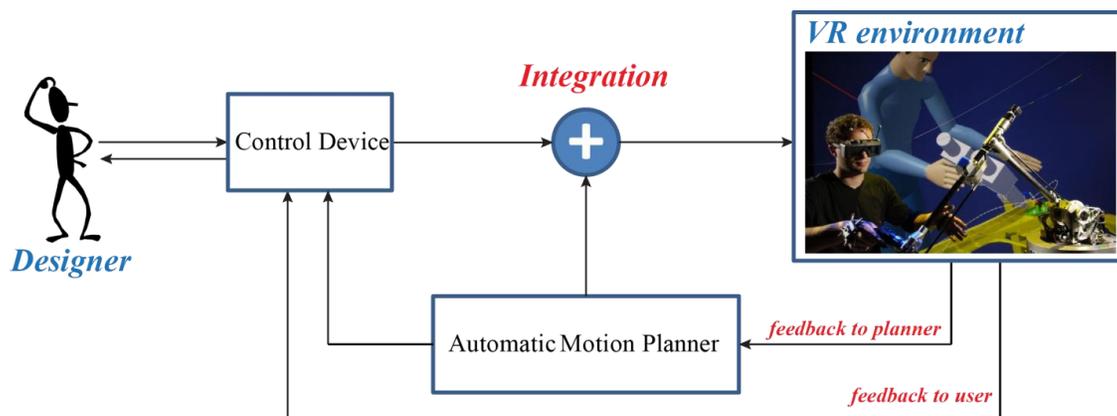


Figure 2.1. Loop integrating designers into the motion planning process

As shown in Fig. 2.1, the interactive motion planning (IMP) process is a closed loop, the human manipulation and automatic motion planning algorithm are combined together to plan the movements of digital models, the planning results from the simulation of digital world provide feedbacks to human and motion planner, in which the former helps human get an intuitive feeling about the current situation and the latter helps improving the local search of motion planner. At the same time, the motion planner also provides a local guidance to human. This is the general framework how the interactive motion planning method works. However, both of the convenience of human's manipulation and the efficiency of motion planner are dramatically influenced by the adopted strategy of integration. Besides, the narrow passage which is regularly seen in the assembly and disassembly task is the trouble-stone for the interactive motion planning method.

In this chapter, we firstly introduce some previous work in IMP. Then we propose a novel strategy of integration for the interactive motion planning system, which integrates the human manipulation, the random retraction method and the RRT-connect algorithm. A large section is dedicated to the analysis of our random retraction method and some principles of implementing the retraction process, which are necessary for a successful interactive motion planner.

## 2.2. Previous work in the interactive motion planning (IMP)

The earliest work to our knowledge which studied the interactive motion planning system was proposed by (Hwang, Cho et al. 1997). The basic idea of their system is firstly allowing users visually specify some via-configurations or sub-goals with haptic devices. Then the automatic motion planners find collision-free paths between these sub-goals. Similar idea was also implemented on an industrial tele-operated robot by (He and Chen 2009). However, in practical implementation, the via-configurations in complex environment are unlikely to be found by human or impossible to be reached by automatic planners.

(Bayazit, Song et al. 2001) permitted users manipulate the object in virtual environment with haptic device to generate collision-free or approximate collision-free (the moving object is shrunk in interaction) motion paths. Then an iterative process with repeated push and local connection found a valid motion trajectory. However, their pushing method is oversimplified for most situations which limit their application. In spite of this, some excellent principles in this paper such as releasing collision constraints and iterative retraction have been followed and developed into some most successful motion planners to date in industrial products (Ferr é Laumond et al. 2005).

In (Ladeveze, Fourquet et al. 2009), (V ázquez Hurtado and Rosell Gratac òs 2008) and (Guzm án, Berenguel et al. 2008), a preliminary workspace discretization is firstly implemented based on the decomposition algorithm such as Octree according to the local collision information, and then a corridor between initial configuration and goal configuration is found. The sampling based motion planners explore in this corridor to find a feasible motion path. Meanwhile, the harmonic functions are formulated in the corridor to provide force feedback. The pre-computed corridor reduces the searching space of sampling based motion planners and thus greatly accelerates the exploration. However, the side-effect of corridor limits the freedom of performance by user, which is highly possible to raise conflicts to user's preference. The evaluation of the ergonomic subjects on these corridors is extremely difficult.

Method proposed in (Patil and Alterovitz 2010) is successfully applied to interactive motion planning for steerable needles. The kinetic model of the plastic steerable needle was constructed

prior to the planning phase. Based on the kinetic model, the needle deforms in real time with the user's manipulation. Although this method is difficult to be implemented to other kind of moving objects such as the non-convex assembly object, it reveals an interesting direction that if we can model a motion policy for the moving object in local environment, the complex manipulation of human may be released to only translation which is much easier for user, which partially inspired our motion learning method in next chapter.

In the methods stated above, the searches of automatic algorithm and human's manipulation are comparatively regarded as two separated stages. A substantial method which allows simultaneously cooperation between the automatic algorithm and user was proposed by (Ta ě, Flavign é et al. 2012). As the roadmap tree of automatic algorithm iteratively enlarges, this method uses the directions of motion controlled by the user to favor the directions of search extensions. At the same time, the planning algorithm provides useful information to the user concerning environment, obstacles and exploration status, which is represented as force feedback guiding user's manipulations. Although this method has been successfully used to the path planning of locomotion animation (Flavign é and Ta ě 2011), it faces great challenges in design filed where the narrow passages are inevitable. In narrow passages, both of the manual manipulation by human and the search by automatic algorithm get into trap and are unable to favor each other.

The narrow passage becomes a bottleneck at present which limits the improvement of the IMP. Moreover, all of the IMP methods rely heavily on the haptic feedback in the interaction with user. Consequently, some of more natural interactive devices such as the infrared motion tracking system and Microsoft Kinect, which provide convenient tools for analyzing ergonomic issues of whole body movements in product design process, are still difficult to implement in IMP. The deeper integration strategy which can reduce the impact of narrow passage is essential for a more sophisticated IMP method.

In this chapter, we propose a novel IMP strategy in (Yan, Poirson et al. 2013, Yan, Poirson et al. 2013), based on the fact that it is easier for user to find a solution in the case of a free or low-constrained environment. The design of our integration strategy aims to relieve user of burdensome manipulations and accelerate exploration in narrow passage. Specifically, we relax the collision check in the interaction process, which allows user to manipulate digital object with some penetration into the obstacle. Then the in-collision configurations sampled in the motion trajectory generated by user are retracted to collision-free. After retraction, we use RRT-connect algorithm to connect every neighboring two seeds to get a valid motion trajectory. The brief comparison between our method and other IMP methods are provided in Tab 2.1. Ideas similar to some components of our IMP method may appear in other work in the literature. However, the combination of them all by our strategy is quite new. For example, although the idea adding retraction into automatic

algorithms has been intensely studied, we not only propose a novel retraction method and integrate it with automatic algorithm but also define some clear principles for implementing the retraction method in IMP application.

Table 2.1. Comparison between different interactive motion planning algorithm

IMPs	Highlights	Year
Bayazit et al.	1.Human generate original path (Object shrunk)	2001
	2.Simply push in-collision configurations out to collision-free	
	3.Add new configurations to roadmap of PRM and search a new path	
	4.Iteratively push and connect until a valid path is found	
He and Chen	1.Human specify Via-configurations or sub-goals	2009
	2.The search of PRM is guided by via configurations or sub-goals	
Vazquez et al.	1.Workspace decomposition	2008
Ladeveze et al.	2.Searching a corridor in workspace with A* algorithm	2009
Guzmán et al.	3.Automatic planners search in corridor and provide force feedback to human	
Patil and Alterovitz	1.Human and computer work synchronously	2010
	2.Searching by sampling based algorithm, local connecting with kinetic model	
Ta ě et al.	1.Human and computer work synchronously	2012
	2.Configurations searched by human add to RRT searching tree;	
	3.RRT searching tree attracts human’s manipulation by force feedback	
Our method	1.Human generate original path (Collision relaxed)	2013
	2.Random Retraction method retract in-collision samples into collision-free seeds on contact space	
	3.RRT-connect algorithm search between every neighboring two seeds to find a feasible path	

## 2.3. Retraction based on the penetration depth

In our interactive motion planning method, the digital model manipulated by the user is allowed to have some penetration into the obstacles. Some parts of the motion trajectory created by the user may be in-collision. The proportion of the in-collision parts will increase when the local environment is narrower and more constrained. In implementation, we take a sparse sample from the recorded trajectory. If any configuration in the picked up samples are in-collision, we use our random retraction method to retract it to the configuration in the nearby collision-free space or the Contact-space ( $\mathcal{CS}_{contact}$ ). Using the retraction strategy to enhance the sampling of automatic algorithms has been intensely studied, which shows good quality on improving the search of

automatic algorithms in the narrow passage. In this section, we will firstly make a brief investigation of these techniques, then introduce our random retraction method and analyze its robustness.

### 2.3.1. Retraction based motion planner

The work based on the retraction intends to design a strategy which has low cost computation and use the information of in-collision configurations to serve the exploration in the collision-free configuration space ( $\mathcal{CS}_{free}$ ). Since it is easier for the sampling based algorithms to find a solution in the low-constrained environment, some retraction based planners dilate the  $\mathcal{CS}_{free}$  through shrinking the obstacles. In (Ferre and Laumond 2004), the algorithm starts from a rough solution found by RRT algorithm in the condition of obstacles shrunk, and iteratively refines it by relaxing the shrinking. Compared to searching a collision-free configuration around the in-collision configuration, the high cost computation is avoided by controlling the deformation of CAD models, and the robustness is guaranteed at the same time. However, the solutions found in several initial steps may be apparently far from the final path, because the topological structure of the shrunk object may be quite different.

The idea of slightly shrinking obstacles is also implemented in some other methods. However, different from the previous method, these methods all try to directly retract the in-collision configurations to collision-free by some simple heuristics. Given an in-collision sample ( $q_r$ ), (Saha, Latombe et al. 2005, Hsu, Sánchez-Ante et al. 2006) retracts  $q_r$  by sampling configurations inside balls of increasing radius centered at  $q_r$ . The radius slightly increases until a collision-free configuration is found or the maximum number of iterations has been reached. (Erbes 2013) also retracts  $q_r$  by an iterative process with small repulsive steps. The translational and rotational momentums of every step are computed based on the normal vectors of the collision points which lie on the contact surface of two interpenetrating CAD objects.

The clear definition of retraction process was firstly provided by (Zhang and Manocha 2008). They indicated that computing the closest boundary point for an in-collision configuration boils down to the computation of the generalized penetration depth. For the given configuration ( $q_r$ ), the retraction step has been formulated as an optimization process which searches the closest point  $q_m$  to  $q_r$  on the  $\mathcal{CS}_{contact}$  based on an appropriate distance metric. In implementation, they firstly search a rough configuration  $q_c$  on  $\mathcal{CS}_{contact}$  and iteratively approach the  $q_m$  along the local contact space formed by the closest feature pairs. Due to the underlying rigorous formulation of generalized penetration depth, their method has been successfully applied to the path planning of disassembly part (Zhang, Huang et al. 2008, Lee, Kwon et al. 2012) and articulated model (Pan, Zhang et al. 2010).

The retraction based methods mentioned above aim to enhance the expansion of search tree by increasing the successful samples in the narrow passage. To our knowledge, none of the previous works has integrated the retraction method into an interactive motion planning process. In our situation, searching the closest point on  $\mathcal{CS}_{contact}$  actually is not necessary, but we do need retract the human manipulated in-collision configuration to collision-free in a short time and the retracted collision-free configuration must comply with human's manipulation trajectory, meaning the generated motion path from these retracted configurations should not have dramatic deformation compared to human's manipulation trajectory. However, this consistency is not guaranteed in the retraction based methods mentioned above. Hence, we propose a novel retraction strategy in (Yan, Poirson et al. 2013, Yan, Poirson et al. 2013) for our IMP method. On each retraction step, our method randomly selects a descending direction based on the appropriate estimate of penetration between two objects. An intuitive comparison between different retraction methods is shown in Fig. 2.2.

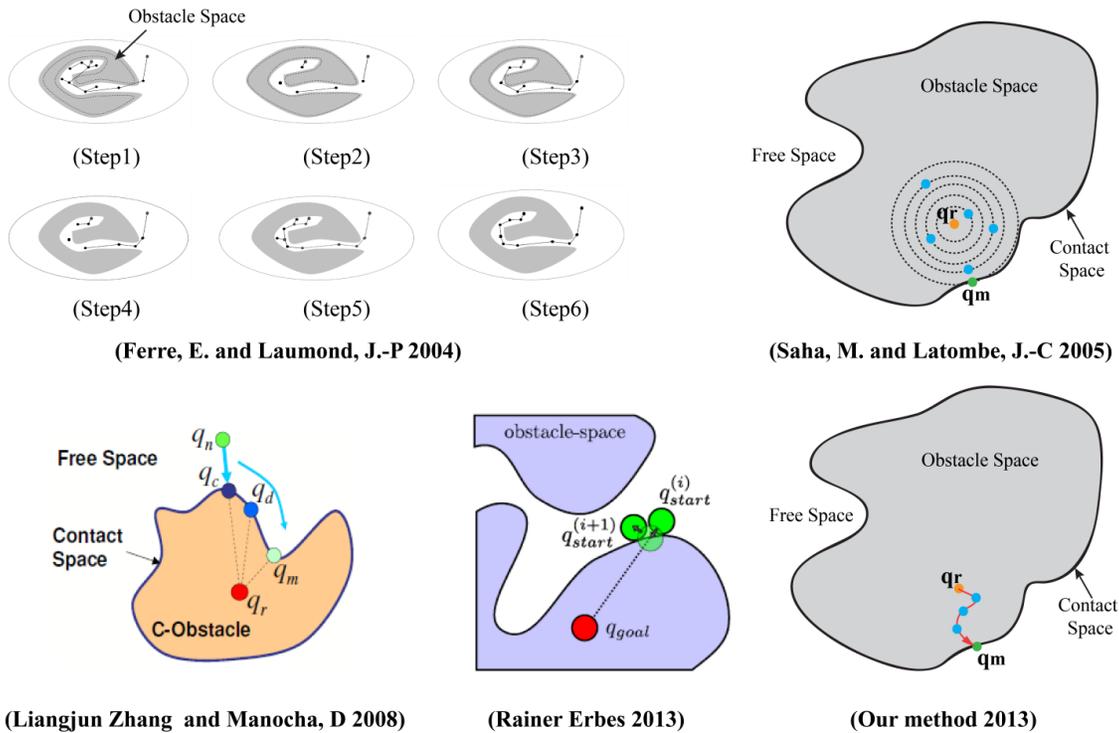


Figure 2.2. Comparison of retraction processes in different methods

### 2.3.2. Random retraction method

Integrating the retraction method into the interactive motion planning process has to consider three rules in design: low cost, consistency and constraints of manipulation. The low cost means that the computation of retraction should be as fast as possible. The rigorous retraction is computing the closest point on  $\mathcal{CS}_{contact}$  for an in-collision configuration. The C-contact space  $\mathcal{CS}_{contact}$  is the subset of configuration-space which consists of the configurations when the object touches one or more obstacles without any penetration. The computation of C-contact space is complex and time-consuming at present, hence (Zhang, Kim et al. 2007) search the closest point along the local simulated contact surface according to the collision features. However, in our application, the ‘closest’ requirement is not very necessary. As shown in Fig. 2.3 (a), for a given in-collision configuration ( $q_r$ ), the generalized penetration depth (Zhang, Kim et al. 2007) tries to find the optimal configuration  $q_m$  on C-contact space. In contrast, our random retraction method tries to find a local optimal configuration  $q_m'$  on C-contact space in the nearby region of ( $q_r$ ).

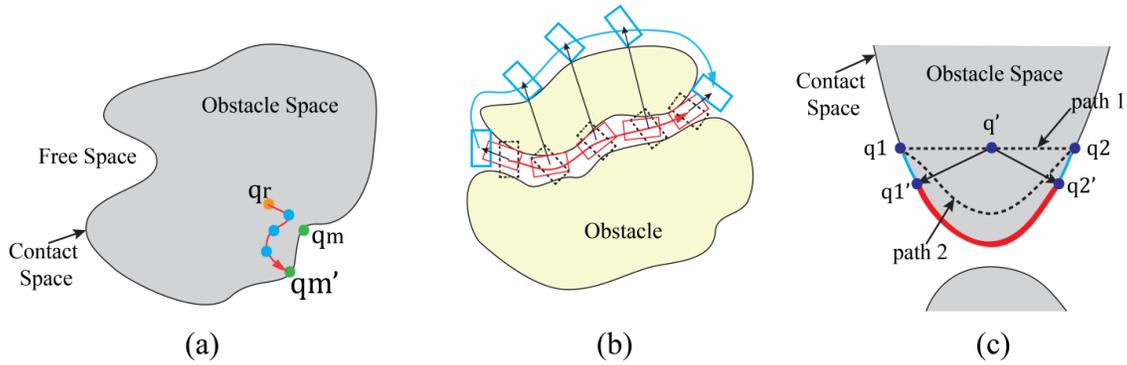


Figure 2.3. The cases related to the three rules considered in our retraction design

The second rule we have to consider in design is the consistency, which means the final path generated from the retracted configurations should follow the similar direction or lie in the same passage compared to human’s manipulation trajectory. As shown in Fig. 2.3 (b) for an example, some states of the rectangle object picked up from the human’s manipulation trajectory are shown as dashed line. In these states, the object is in-collision with the two obstacles. The translational penetration depth ( $PD'$ ) computation method will retract these configurations out of the narrow passage and far from the original trajectory (Kim, Otaduy et al. 2003, Je, Tang et al. 2012), which is shown as blue color in Fig. 2.3 (b). However, the retracted configurations that we need should not deviate dramatically from the original directions and regions searched by humans. The appropriate retracted states are shown as red color.

The third rule is related to the second rule, because the consistency is possible to guarantee only if we define some clear constraints to the manipulation mode of human. And this problem is also related to the continuous issue of the penetration depth computation (Zhang, Kim et al. 2014). If no constraint is defined to user's manipulation, it is quite possible that user could generate a straight path from  $q_1$  to  $q_2$  shown as 'path 1' in Fig. 2.3 (c). The configuration  $q'$  is the path's middle point. Except (Ferre and Laumond 2004), all of the retraction methods are designed to retract the in-collision configuration with minimal transformation cost. As a result, the configurations lie in path between  $q_1$  and  $q'$  will be retracted to configurations on C-contact space between  $q_1$  and  $q'_1$ . The treatment to configurations between  $q'$  and  $q_2$  is the same. However, the final path we need is a continuous path from  $q_1$  to  $q_2$  which lies on C-contact space shown as the red curve. In our application, the original rough trajectory created by the user is similar to the situation of 'path 2' which guarantees the consistency issue.

In considering the three rules and the requirement of relieving user from burdensome manual manipulation, we proposed our random retraction method in. Following the formalized definition of generalized penetration depth, we also describe our retraction process as an optimization problem. Starting from an in-collision configuration  $q_r$ , our retraction method searches a configuration  $q_m$  in interior or on the boundary of C-obstacle space with minimal penetration into obstacles, which can be formally defined as:

$$q_m = \arg \min_q \delta(q, obstacles), q \in CS_{obstacle} \quad (2.1)$$

In which,  $\delta$  is a metric which estimates the quantity of penetration into obstacles when the object is at configuration  $q$ . The retraction is successful when the configuration  $q_m$  is on the C-contact space or the collision-free space and in the nearby region of the configuration  $q_r$ .

According to the three rules, the computation of metric  $\delta$  should be low cost and have no dependency with the object's translational and rotational movements in configuration space, which will be discussed in the next section. We carefully design the retraction strategy which starts from the in-collision configuration  $q_r$ , in order to make the final configuration  $q_m$  stays in the local nearby region of configuration  $q_r$ . As the computation of C-obstacle space  $CS_{obstacle}$  and the steepest decent direction of  $\delta$  are complex and time-consuming at present, the conventional optimization algorithms are either impossible to implementation or high cost. Therefore, instead of using the mesh grid as in the pattern search optimization algorithm, we take a random direction  $D_{random}$  at each iteration step. If the sampled direction or its reverse direction is a decent direction, we retract the object along the decent direction with the step length  $SL$  and enlarge the  $SL$  with proportion  $\rho > 1$ . Otherwise, if the two directions fail, the step length  $SL$  is decreased by proportion  $\gamma < 1$ . The step length  $SL$  is also set beneath a threshold  $SL_{max}$  in order to avoid the final

configuration locating far from the nearby region of  $q_r$ . The iterations stop when the configuration after step retraction is collision-free or the  $\delta$  is less than a threshold  $\delta < \varepsilon$ . According to the collision-free configuration, the configuration  $q_m$  can be easily found out by the binary search. The variable step length contributes to not only accelerate the convergence speed but also guarantee the algorithm can find a local optimal point. The magnitude of  $\varepsilon$  is set very small to stop the iteration when there is no solution in the nearby region around the configuration  $q_r$ . And the algorithm always converges to the local optimal point before the  $\delta$  is less than  $\varepsilon$ . The pseudo code of the algorithm is shown below.

---

**Algorithm 2.1:** Random Retraction Method (RR)
 

---

```

1  input:  $q = q_r$ , Step length  $SL$ , Obstacles;
2   $\delta \leftarrow Intersection\_estimate(q, Obstacles)$ 
3  while  $\delta > \varepsilon$  and  $collision(q) = true$  do
4       $D_{random} \leftarrow Random\_direction()$ 
5       $q1 \leftarrow move(q, SL, D_{random})$ 
6       $q2 \leftarrow move(q, SL, \sim D_{random})$ 
7       $\delta1 \leftarrow Intersection\_estimate(q1, Obstacles)$ 
8       $\delta2 \leftarrow Intersection\_estimate(q2, Obstacles)$ 
9      if  $\delta1 < \delta$  then
10          $q \leftarrow move(q, SL, D_{random})$ 
11          $\delta \leftarrow \delta1$ 
12          $SL \leftarrow SL * \rho$  ( $SL < SL_{max}$ )
13     else if  $\delta2 < \delta$  then
14          $q \leftarrow move(q, SL, \sim D_{random})$ 
15          $\delta \leftarrow \delta2$ 
16          $SL \leftarrow SL * \rho$  ( $SL < SL_{max}$ )
17     else
18          $SL = SL * \gamma$ 
19     end if
20 end while

```

---

### 2.3.3. Estimation of the penetration

The metric  $\delta$  estimates the penetration between the object and obstacles, which shouldn't depend on the translational or rotational movements of the object, otherwise, the problem breaking the

consistency rule as the situation in Fig. 2.3 (b) is quite possible to happen. In our application, the penetration is estimated by calculating two objects' interpenetrating area in 2D and the interpenetrating volume in 3D. In Fig. 2.4 (a), we show a situation in 2D, in which the object controlled by user has a valid penetration into the obstacle. In 2D, the interpenetrating area can be easily calculated by the Boolean operation between polygons. In 3D situations, to our knowledge, we haven't found an exact algorithm which calculates the interpenetrating volume between two objects. Instead, the conventional methods decompose the CAD model with hierarchical sphere trees and measure the intersecting volume by counting the total interpenetrating volume between different spheres (Weller and Zachmann 2009, Weller and Zachmann 2011) and (Jacquenot 2010). The sphere tree constructing method in (Jacquenot 2010) may change the topology of complex object, hence the conception of (Weller and Zachmann 2009) which fills the interior of CAD model with spheres is more suitable for our application in 3D. In Fig. 2.4 (b), the centers of interpenetrating spheres between two CAD models are marked as red points.

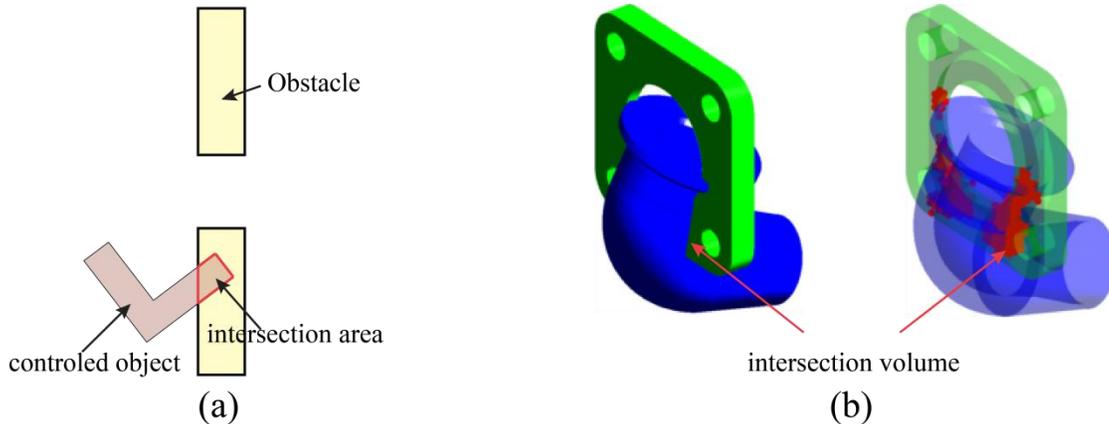


Figure 2.4. Penetration estimation in 2D and 3D

In our application, the spheres filling the interior of CAD model are constructed through a voxelization process. After importing the CAD file, a bounding box  $(I_x, I_y, I_z)$  of the object can be constructed according to the position and orientation of the object, and  $I_x, I_y, I_z$  are the lengths of the bounding box along the  $x, y, z$  dimensions. Supposing  $I_{max}$  is the maximum value of  $I_x, I_y$  and  $I_z$ , the resolution  $R$  of the voxel enclosed by the bounding box is  $R = I_{max}/N$  in which  $N$  is a predefined parameter. The voxel of the bounding box for the object in Fig. 2.5 (a) is shown in Fig. 2.5 (b). Then the points in the interior of the object are retained and the others are discarded, as shown in Fig. 2.5 (c). These retained points are used as the centers to generate sphere array, as shown in Fig. 2.5 (d). The radius of the sphere is  $R/2$ . The sphere arrays in different resolution are shown in Fig. 2.6.



interpenetrating volume may not be able to correctly reflect the proximity situation when one object is completely inside the other, in which case the estimation of penetration remains constant regardless of the amount of interpenetration (Zhang, Kim et al. 2014). Like the situation in Fig. 2.7 (a), if the controlled object is allowed to be completely inside the obstacle, the human controlled path is impossible to be retracted. Hence, we do not allow the object to penetrate completely into obstacle in the interaction. This limitation can be simply realized by checking if the interpenetrating volume equals to the volume of the controlled object.

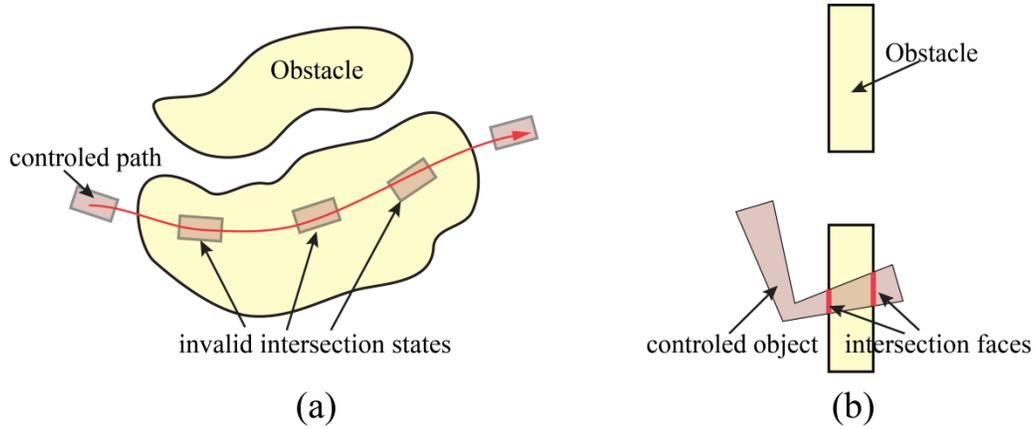


Figure 2.7. Two cases with invalid penetration

Another situation influencing the retraction consistency is that the object may penetrate through the obstacle, in which the decent direction of  $\delta$  depends on the shapes of object and obstacle. As shown in Fig. 2.7 (b), the controlled object penetrates through the obstacle. It's obvious that the local decent direction will drag the object to the configuration with less  $\delta$  but deeper penetration. In order to avoid this situation, at most one penetration surface is allowed between the controlled object and each obstacle. In other words, the penetration surfaces on each obstacle must be the connected domain. The penetration surface can be simply detected in real time by the polygon operations in 2D and collision-detection algorithm in 3D, such as PQP (Larsen, Gottschalk et al. 1999).

#### 2.3.4. Convergence

For a given in-collision configuration  $q_r$ , the convergence of retraction means that the algorithm can always find a configuration in the C-contact space which is located in the nearby region of  $q_r$ . In last section, we indicate that the controlled object is not allowed to penetrate completely or through the obstacle in order to guarantee the consistency. The two limitations we defined also guarantee there always exist a decent direction in the local region of configuration  $q_r$ . In a single retraction step, if moving the object along a randomly sampled direction or its reverse direction with the current step length can reduce the  $\delta$ , we consider it as a successful retraction step. The

convergence is guaranteed when all these steps are successful. This is realized by dynamically controlling the step length in our algorithm.

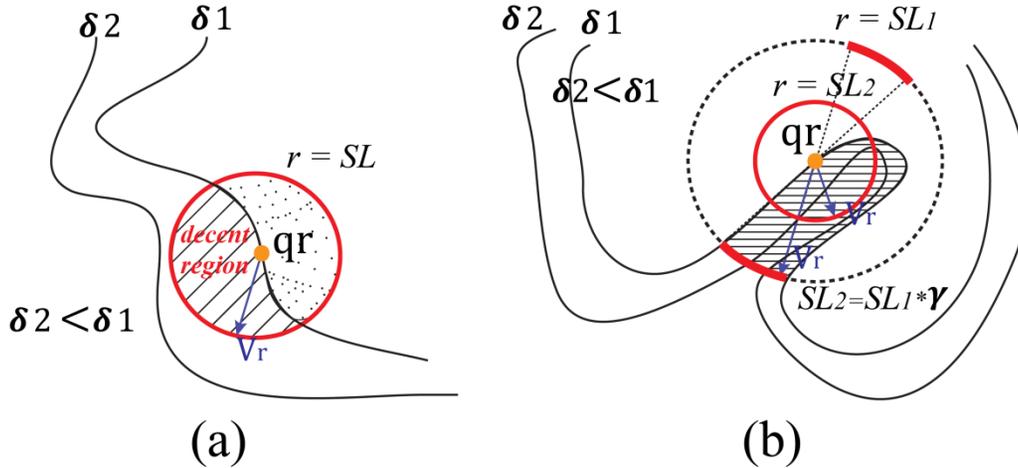


Figure 2.8. The situations of convergence in two different cases

As we adopt the random sampling, the probability of success at each retraction step can be regarded as the percentage of decent vectors and their reverse vectors in a circle which is centered at reference configuration  $q_r$  with radius equaling the current step length. In Fig. 2.8 (a), we show a certain situation in which any sampled direction or its reverse direction is decent. In this situation, the decent trend at reference configuration  $q_r$  is depicted by the contour lines of  $\delta$ . The contour line  $\delta_1$  transverses the sphere enclosed by the cycle centered at  $q_r$  with radius equaling current step length  $SL$ . We can notice that the contour line separates the sphere into two regions in which the decent region is marked by twill texture. No matter the random vector  $V_r$  points to the boundary of decent region or not, a decent direction can be surely found.

In some tough situations, there is poor possibility for the randomly generated vector pointing to a decent direction. Such a situation is shown in Fig. 2.8 (b). The decent trend at reference configuration  $q_r$  is also depicted by the contour lines of  $\delta$ . In condition with the original step length  $SL_1$ , the circle which contains all possible directions is marked by the dashed line. The decent region can project onto a section of arc on the circle. The arc and its reflected arc on the circle constitute the directions which can lead to a decent move. However, this kind of directions takes a small proportion of the circle, which makes it difficult for the randomly generated directions fall in decent region. In this situation, we scale down the circle by decreasing the step length by a proportion  $\gamma < 1$ . The  $\gamma$  is set 0.5 in our application. After scaling, if the generated random vector or its reverse vector is still not able to point to a decent direction, the down scaling will be continued until a situation similar to the certain situation appears.

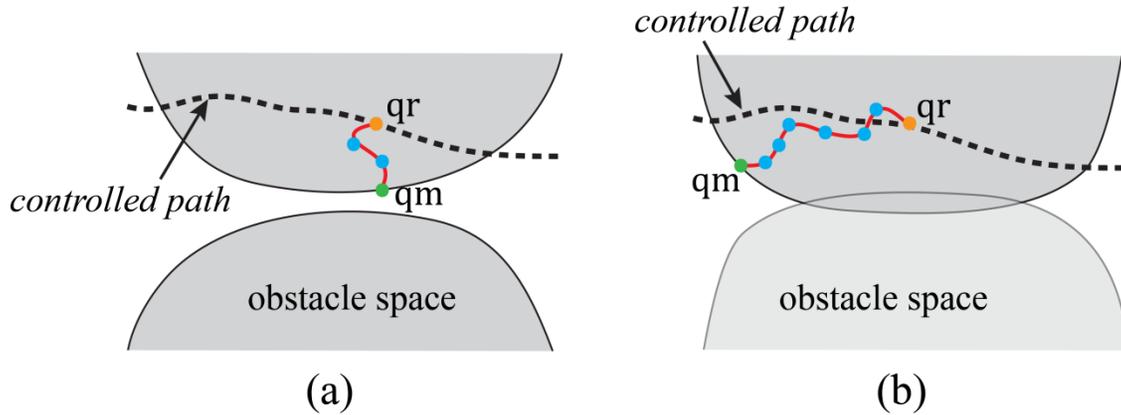


Figure 2.9. The comparison between a ‘good’ situation and a ‘bad’ situation

There are also some situations that we call them as ‘bad’ situations. These situations usually happen when the narrow passage passed though by the controlled object has no accessibility. In a good situation, the in-collision configuration  $q_r$  will be retracted with a few steps onto the C-contact space in the narrow passage, as shown in Fig. 2.9 (a). In contrast, the retraction algorithm will spend much more time to find the target configuration  $q_m$  with a lot of retraction steps in a bad situation, as shown in Fig. 2.9 (b). This will degrade the efficiency of our retraction process for the whole human controlled trajectory. However, we find in experiment that the bad situation happens with a low frequency, because human always try another passage which seems easier for his manipulation before the controlled object passes through the non-feasible passage.

### 2.3.5. Retraction results in applications

In the following experiments, we applied our random retraction algorithm to a moving non-convex object controlled by user. In this application, the L shape object has to pass through a hole of the obstacle. The task involves a narrow passage which is difficult for both the automatic algorithms and the manual manipulation of human. All the experiments are performed in the software platform Matlab 2013a and on a PC with CPU i5-3317U 1.7GHz and 8 GB ram.

In these experiments, we describe the state of the object with a triple  $q = (x, y, \theta)$ . The translational coordinates indicate the position of the reference point on the object, as shown in Fig. 2.10 (a). The parameter  $\theta$  indicates the angle by which the bottom line of the object rotates counterclockwise. As shown in Fig 2. 10 (a), the controlled object penetrates into the obstacle at configuration  $q_r$ . Our random retraction algorithm retracted the penetrated object to the configuration  $q_m$ , at which the object touches the obstacle and stays in a nearby region of the configuration  $q_r$ . In Fig 2. 10 (b-d), we demonstrate the trace of the retraction process on three planes in the space  $(x, y, \theta, \delta)$ . These planes are chosen by fixing one parameter of the triple  $(x, y, \theta)$  and changing the other two. The

contour lines projected onto these planes are marked by the value of  $\delta$ . Even though we adopt the fastest function in Matlab to calculate the interpenetrating area between two polygons by sparse sampling, the construction of these contour lines is still expensive. Therefore the conventional optimization algorithm may not be able to solve the retraction problem in a reasonable time. In contrast, our random retraction algorithm spends a small cost at each iteration step to find a decent direction. Even though these randomly generated directions may not seem favorable as well as the steepest decent direction, the total speed of retraction process is fast. The time spent in retraction process by our method is shown in Tab. 2.2. Moreover, the convergence of our retraction algorithm is always infallible. In this case, as the controlled object is not allowed penetrating through the obstacle, the configuration  $q_r$  actually reaches the critical condition that some further manipulations may break the limitation. According to the changing trend of contour lines on the three planes, if the object moves along any direction which increases the estimation of penetration  $\delta$ , the object may easily penetrate through the obstacle. However, considering the step length, reaching the decent region on the other side needs the configuration  $q_r$  translates cross the increasing region. Thus the retraction step which may lead the object to a configuration penetrating through the obstacle is not possible to happen. This avoids the non-consistency situations as shown in Fig. 2.3 (b) (c) happen.

In Fig. 2.11, we show a tougher situation in which the object intersects with the obstacle within the tightly constrained tunnel. The allowed penetration greatly reduces the difficulty of manipulation of user in the cluttered environment. Similar to the last case, our random retraction algorithm retracted the penetrated object to the configuration  $q_m$ , at which the object touches the obstacle and stays in a nearby region of the in-collision configuration  $q_r$ . In this case, the optimization process needs to search in a narrow gap of C-obstacle space as shown by the contour lines in Fig. 2. 11 (c) (d). As we can see, our retraction algorithm dynamically decreases the step length to guarantee a decent direction and increases the step length to escape the trap and accelerate the convergence once a decent direction is found. This can also be proved by the results in Tab. 2.3, in which the decreasing speed of  $\delta$  always changes with the step length growing and scaling down. For example, the value of  $\delta$  '0.0102' appears three times in the table, which implies the step length changes three times at that configuration to find a decent direction.

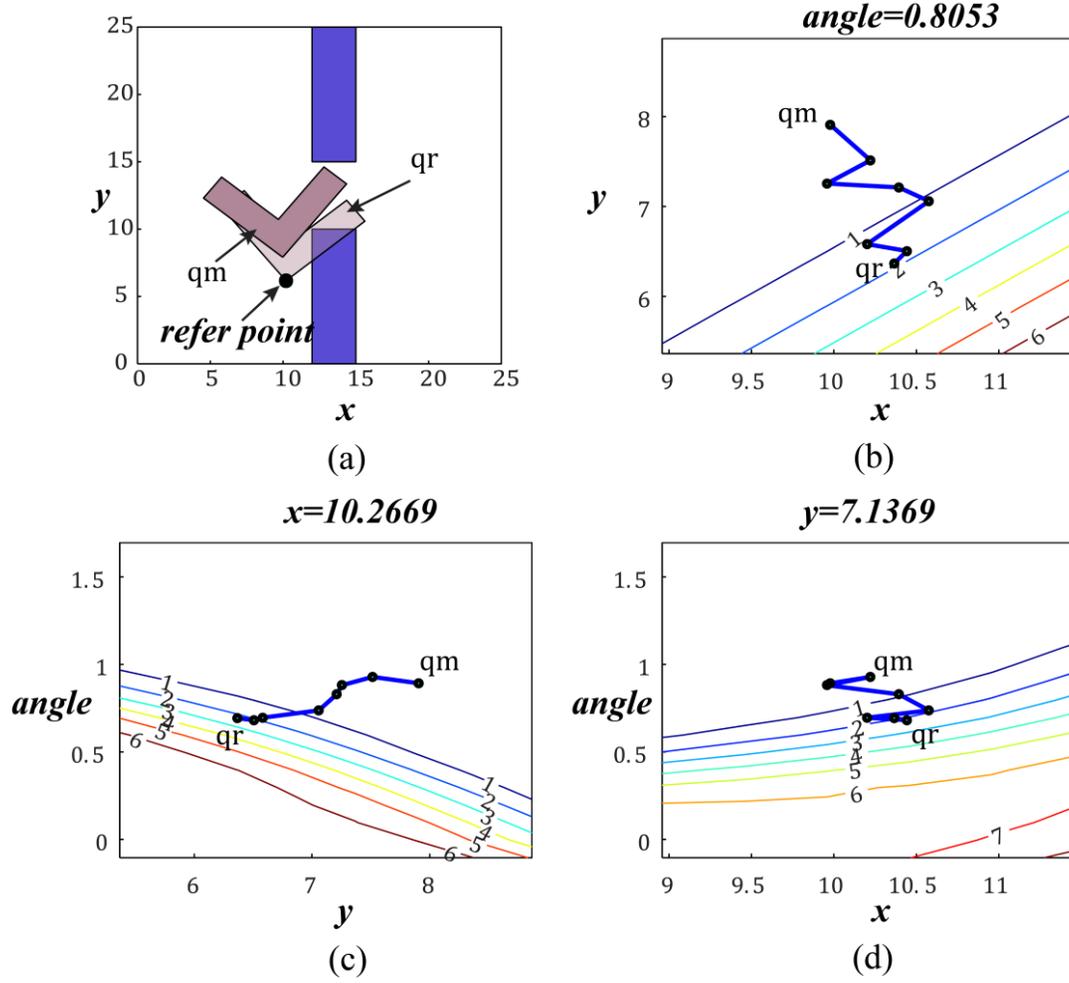


Figure 2.10. Experiment results of random retraction algorithm in the first case

Table 2.2. The spent time and the change of  $\delta$  in the first retraction case

$\delta$	3.1182→3.0662→2.1969→1.4939→0.4909→0.0303→0.0043→0
Retraction time (s)	0.2276
Average time (s)	0.0325

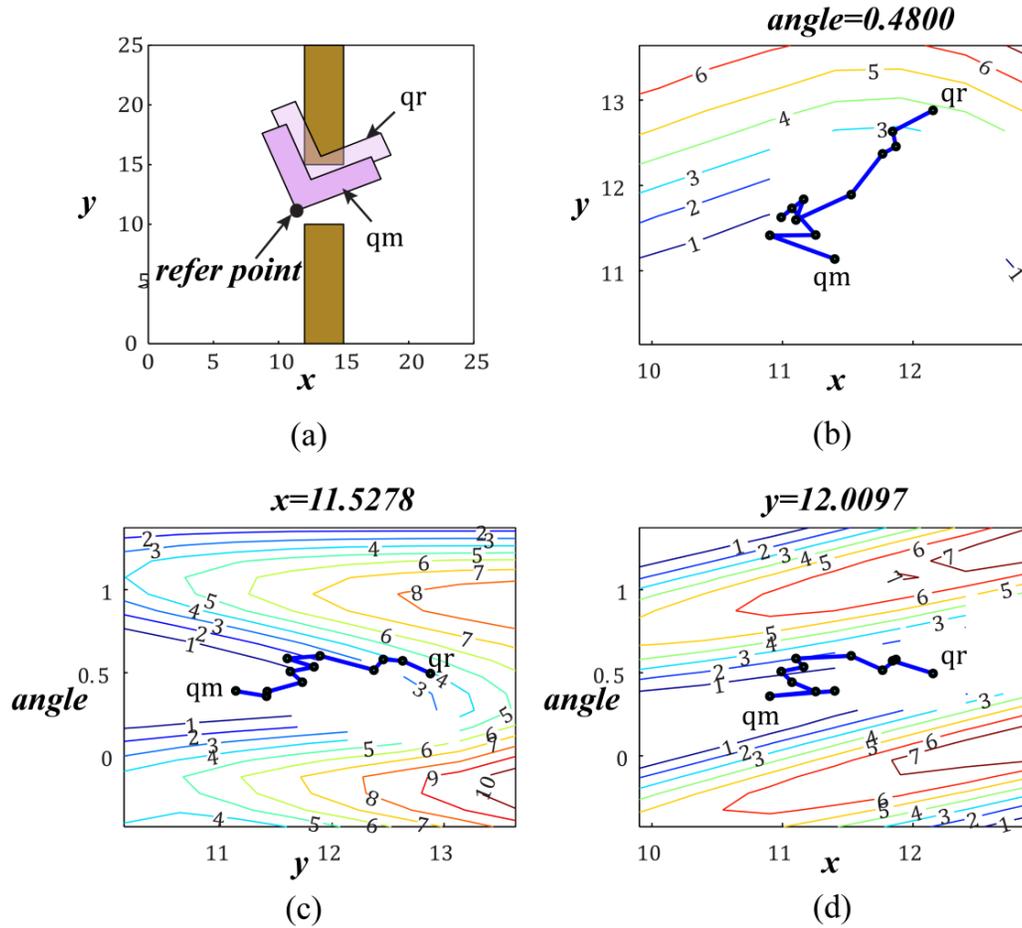


Figure 2.11. Experiment results of random retraction algorithm in the second case

Table 2.3. The spent time and the change of  $\delta$  in the second retraction case

$\delta$	3.7504→3.4372→2.9717→2.2835→2.1151→2.1151→1.8926→1.7548 →1.7548→1.7101→0.7016→0.0126→0.0102→0.0102→0.0102→0
Retraction time (s)	0.3845
Average time (s)	0.0256

In 3D example, the retraction has been applied to the famous flange benchmark which is a motion planning problem containing a narrow passage. This benchmark is the representative of a type of

problem that might occur in a maintainability study of a mechanical CAD design. The environment consists of a fixed rectangular part with a circular opening (the obstacle) and an elbow shaped curved pipe (the object) that must be inserted into the opening in the obstacle. This problem requires generating a sliding motion where the obstacle and object are nearly touching with the object twisted into or out of the obstacle. Finding valid motions in this constrained space is a difficult problem with many approaches developed to find these portions of the space (in courtesy of the Algorithms & Applications Group of Texas A&M University <https://parasol.tamu.edu/dsmft/benchmarks/mp/>).

In this example, the configuration of the object has been represented as a six-tuple  $(x, y, z, \alpha, \beta, \gamma)$ , in which  $(x, y, z)$  describes the position and  $(\alpha, \beta, \gamma)$  describes the orientation. The first image of Fig. 2.12 shows the configuration  $q_r$  at which the object interpenetrates into the obstacle. In the remaining images of Fig. 2.12, the penetration volume marked by the centers of interpenetrating spheres is shown in different resolutions, in which  $N$  is the parameter controlling the resolution of spheres array (see Fig. 2.6). The objective of our retraction method is to get a nearby configuration around the interpenetrating configuration so that the searching tree of automatic algorithm can be guided in the difficult region. Thus the retraction result which is far from the interpenetrating configuration leads to an invalid retraction, otherwise it is valid. The valid and invalid results retracted from the configuration  $q_r$  are shown in the Fig. 2.13. It can be noticed that the invalid retraction results do not provide useful information for the motion plan which is intend to insert the pipe into the opening of the obstacle.

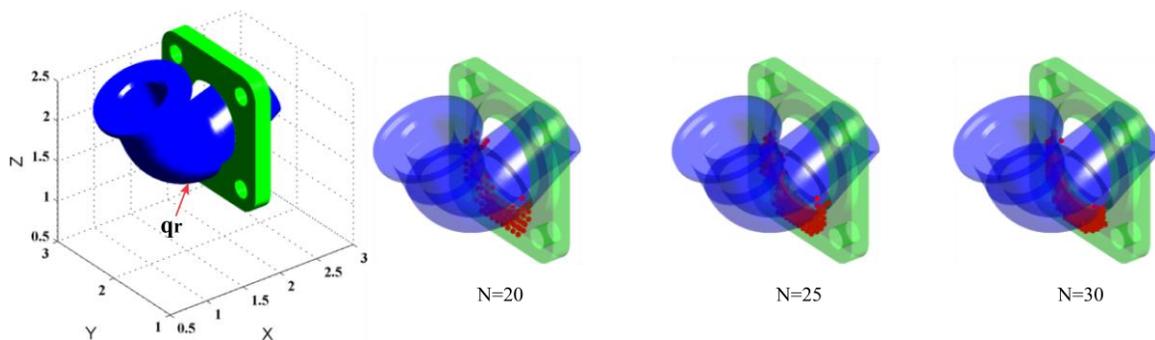


Figure 2.12. Penetration volume in different resolution

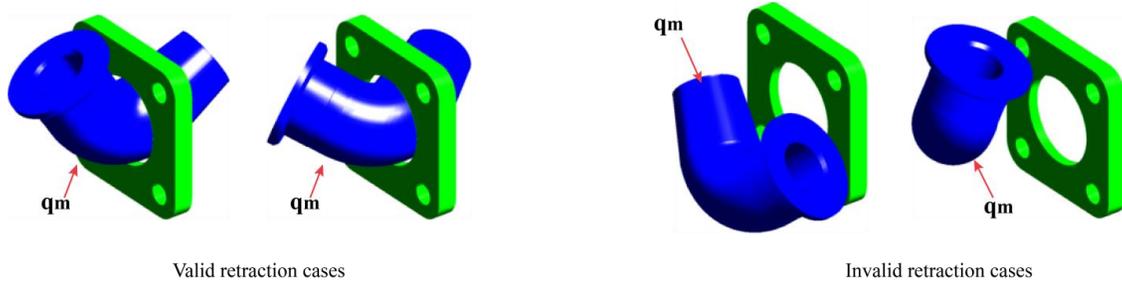


Figure 2.13. Valid and invalid retraction cases

An experiment has been carried out to measure the influence of variable step length and resolution to the retraction process. Intuitively, the smaller the step length is, the more retraction steps our random retraction method needs; and the bigger the resolution parameter  $N$  is, the more time the penetration evaluation spends. Thus, it is hoped that the successful retractions can be realized with the bigger step length  $SL$  and the smaller resolution parameter  $N$ . The value of  $N$  is set as  $N \in \{20, 25, 30\}$  in the experiment to avoid the inaccurate representation of the CAD model and the time-consuming evaluation of penetration. The value of  $SL$  is set as  $SL \in \{0.05, 0.1, 0.20\}$  by user experience. For every group of  $SL$  and  $N$  in  $\{0.05, 0.1, 0.20\} \times \{20, 25, 30\}$ , ten times retractions have been applied to the case of Fig. 2.12. The spent time and rates of successful retraction are shown in the Tab. 2.4 and the number of retraction steps is shown in the Tab. 2.5. The results of these two tables are also illustrated in the Fig. 2.14. From the results of Tab. 2.4 and Fig. 2.14 left, it is obviously noticed that with the enlargement of parameter  $N$ , the average retraction time has significantly increased. Under the same resolution of sphere array, when the step length is enlarged, the average retraction time decreases, whereas the rate of successful retractions descends. The time of a single retraction is decided by two factors: number of retraction steps and time of penetration evaluation. The time of retraction will increase if any one of the two factors is increased. Under the same resolution of sphere array, the time of penetration evaluation remains constant, but the number of retraction steps decreases which makes the retraction time decrease. With the increase of parameter  $N$ , the number of retraction steps has not significant monotonic change, but the time of penetration evaluation greatly increases (see Tab. 2.6), thus the retraction time increases. Although the average retraction time and number of retraction steps show obvious change with respect to the change of parameter  $N$  and step length  $SL$ , the retraction time and number of retraction steps in a single retraction may vary dramatically under the same condition, because the randomly generated retracting direction may approximate or be far from the steepest descent direction.

Table 2.4. The spent time and rates of successful retraction in different resolution and step length

SL	N	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Ave	Min	Max	P
0.05	20	4.456	4.700	2.717	2.610	2.299	1.818	3.548	5.036	4.220	1.910	3.331	1.818	5.036	1
0.10	20	4.030	2.065	2.269	2.439	2.487	1.521	0.809	2.006	5.282	3.150	2.606	0.809	5.282	0.9
0.20	20	1.072	7.080	1.205	1.593	2.257	1.404	3.332	2.114	1.402	0.830	2.229	0.830	7.080	0.3
0.05	25	3.344	2.867	1.759	2.628	5.264	11.598	1.290	3.733	2.892	3.905	3.928	1.290	11.598	1
0.10	25	1.921	1.817	3.990	1.483	2.815	5.256	2.998	2.211	2.163	3.502	2.816	1.483	5.256	1
0.20	25	3.825	0.594	3.632	1.721	1.709	1.851	2.314	3.809	4.630	2.114	2.623	0.594	4.630	0.9
0.05	30	4.556	10.351	5.109	2.473	2.893	5.729	8.213	11.270	6.824	4.130	6.155	2.473	11.270	1
0.10	30	4.403	3.235	7.893	4.247	2.135	3.320	5.930	3.304	3.558	2.888	4.091	2.135	7.893	0.9
0.20	30	5.657	2.808	3.765	3.539	0.944	4.038	3.978	6.520	5.531	4.922	4.170	0.944	6.520	0.8

SL = step length; N = resolution parameter; T1~T10 (s) = time of 10 retractions; Ave = average; Min = minimum; Max = Maximum; P = percentage of valid retractions in 10 retractions.

Table 2.5. The number of retraction steps in different resolution and step length

SL	N	$N_{SL}^1$	$N_{SL}^2$	$N_{SL}^3$	$N_{SL}^4$	$N_{SL}^5$	$N_{SL}^6$	$N_{SL}^7$	$N_{SL}^8$	$N_{SL}^9$	$N_{SL}^{10}$	Ave	Min	Max
0.05	20	37	28	23	53	38	30	23	67	20	94	41.3	20	94
0.10	20	15	21	19	14	29	33	27	40	28	45	27.1	14	45
0.20	20	28	7	16	62	35	38	9	39	31	12	27.7	7	62
0.05	25	29	15	36	39	41	17	17	15	16	42	26.7	15	42
0.10	25	15	59	13	17	24	43	22	11	32	14	25.0	11	59
0.20	25	35	29	21	21	16	22	4	14	32	4	19.8	4	35
0.05	30	18	20	38	26	63	11	29	26	38	17	28.6	11	63
0.10	30	25	28	30	33	25	12	31	48	20	37	28.9	12	48
0.20	30	4	36	19	8	25	18	27	33	21	20	21.1	4	36

SL = step length; N = resolution parameter;  $N_{SL}^1 \sim N_{SL}^{10}$  (s) = number of retraction steps in 10 times retractions; Ave = average; Min = minimum; Max = Maximum.

Table 2.6. The spent time of penetration evaluation in different resolution

	N=20	N=25	N=30
Spheres in Object	884	1644	2733
Spheres in obstacle	384	852	1292
Penetration evaluation (s)	0.0130	0.0456	0.0898

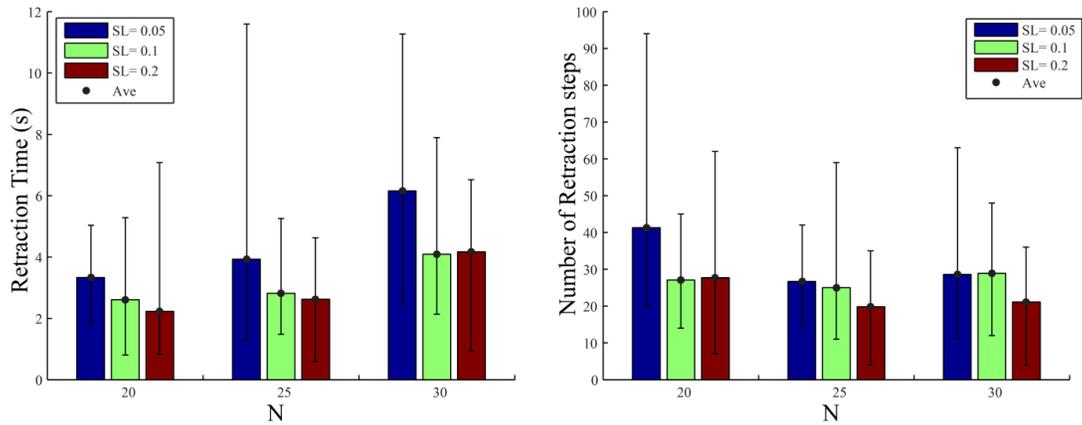


Figure 2.14. The comparison of retraction time and number of retraction steps in different resolution and step length

### 2.3.6. Conclusion

According to the three rules on designing a valid retraction method for an interactive motion planning system, the random retraction method is proposed in this section. Some constraints of manipulation are also defined in order to guarantee the robustness and consistency property of the retraction method. Our retraction method has the good quality of converging to a local configuration on the C-contact space which lies in the nearby region of the in-collision configuration. The convergence is guaranteed in most situations by the variable step length which also influences the converging speed of the retraction. The efficiency and robustness of our retraction algorithm have been analyzed in the challenging cases in 2D and 3D. It will show in the next section how these retracted configurations can efficiently guide the exploration process of the automatic motion planning algorithm in the narrow passages.

There still exist some problems in our retraction algorithm such as the ‘bad’ situations mentioned in section 2.3.4, in which the retraction algorithm usually spends much more time. However, the happening frequency of these situations in practical implementation is low. Moreover, we find in practice that the behaviors in the retraction have some similarities when the penetration happens on the similar part of the object. This reveals the integrated learning mechanism may improve the performance of the retraction process. The similar idea also appears in a recent work of (Pan, Zhang et al. 2013).

## 2.4. IMP based on the retraction and the local connection

Based on the random retraction method analyzed in last section, the in-collision configurations of the human manipulation trajectory can be efficiently retracted to the collision-free configurations or the configurations which only touch the obstacles. The penetration allowed in interaction process

releases the human from burdensome manipulations in tightly constrained environment. The following contents provide more detailed information about our interactive motion planning system, which integrates human's manipulation, the random retraction method and sampling based motion planning algorithm.

### 2.4.1. System global view

The global view of our interactive motion planning framework is presented in Fig. 2.15. The more detailed description of the process is presented in Fig. 2.16. Supposing that the path marked by green points is the user manipulated path. This rough path is then retracted to produce seeds (marked blue) on C-contact space by our Random Retraction Method. After that, BiRRT (RRT-connect) algorithm explores between every neighboring two seeds to find a connecting path. The algorithm either finds a solution from  $q_1$  to  $q_8$  shown as Fig. 2.16 (a), or return the nearest configuration pair (marked red) shown as Fig. 2.16 (b).

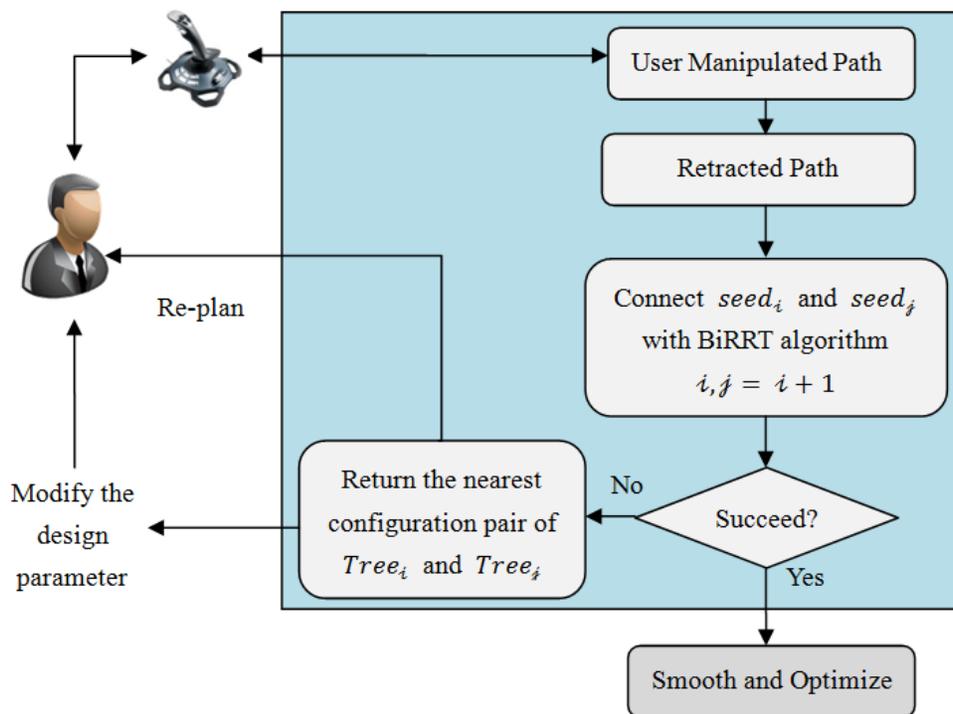


Figure 2.15. Interactive motion planning system

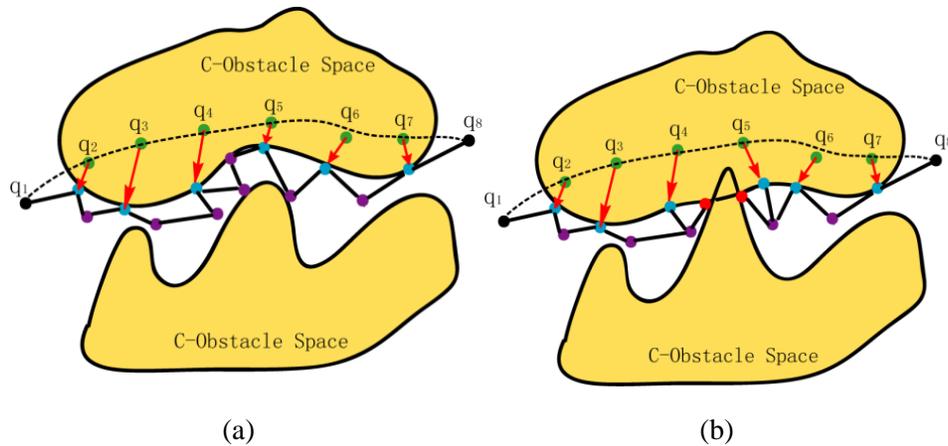


Figure 2.16. Interactive planning in successful case and failed case

Based on the fact that users can effectively finish the motion planning task in a loose space, we firstly allow user manipulate CAD parts with some penetration into obstacles. In this manner, user will quickly find a rough path which contains some in-collision configurations and lies closely to the boundary of C-obstacle space. At the same time, the rough path satisfies some of the user's operating habits and preferences. Such issues are pretty important for the design of an interactive motion planning system applied in product design, because the user's perceptual or cognitive response are more and more considered in the early stage of design circle. For example, in the maintenance study of repairing a car engine, there exist many possible passages which the repairing tool can pass through. However, it is extremely difficult to find the most convenient path only by automatic algorithm, because human's preferences are still difficult to accurately model.

In the interaction process, users can choose different kinds of interactive controlling devices, no matter using the haptic devices or the non-haptic motion-sensing system. In fact, even though the dependency of the haptic response in our application is reduced, the haptic devices are still more suitable for some complex manipulations, because the accurate operations in these situations are inevitable even in condition of released collision checking. In implementation, we take a strategy that raises the sampling rate of human's manipulation trajectory when user's operations encounter collisions more frequently and decreases it in opposite situation. This strategy will increase computational efficiency as well as guarantee a perfect reflection for human's operations in narrow passages.

After retraction, the user manipulated path is transformed into seeds which consist of some collision-free configurations and some configurations on C-contact space. The following process is similar to the multi-tree RRT. We use BiRRT algorithm to connect every neighboring two seeds. Since the neighboring two seeds are close to each other, it's pretty easy for BiRRT planner to find a

path between them. In case of that the motion planner fails to find a solution within a limited time, we return the nearest configuration-pair in workspace. The interference information about which surfaces of the designed CAD model block the accessibility is easy to be detected through analyzing the contact information between the nearest configuration-pair and the obstacles. The user can make movement adjustments or modify the design size of parts according to the interference information.

#### 2.4.2. Seeds connection by the RRT-connect algorithm

The retraction process in our system can be regarded as a repairing process that pushes the samples of human's manipulated trajectory out of the C-obstacle space. After retraction, we can get a sequence of seeds that roughly describe the trend of the final trajectory. We call them 'seeds', because they can be regarded as the milestones that serve to guide the motion planners in local space. Supposing we have a sequence of seeds  $\mathcal{S} = \{Seed_i | i \in [1, n], Seed_i \in \mathcal{CS}_{free} \cup \mathcal{CS}_{contact}\}$ , the mission of local connection module is to find a valid path between every neighboring two seeds. In practice, we find the RRT-connect (BiRRT) algorithm (Kuffner and LaValle 2000) is an appropriate tool for our local connection mission. The RRT-connect algorithm is a variant of RRT algorithm, in which two trees grow respectively from initial and goal configurations to explore and capture the connectivity of C-free space. A greedy heuristic is implemented in this algorithm to make the two growing trees attract each other when expanding. In our application, we set every pair of neighboring two seeds as the initial and goal configurations of RRT-connect algorithm, as shown in Fig. 2.17. Therefore, the connecting process is like running multiple independent RRT-connect connections, which is suitable for parallel processing. The local connection is successful when the two trees growing respectively from the neighboring two seeds are connected in limited time. When the connection is failed, we return a pair of nodes located respectively on the two trees which have the nearest distance among any other pair of nodes. Supposing that the two trees  $\mathcal{T}_i$  and  $\mathcal{T}_{i+1}$  grown from  $Seed_i$  and  $Seed_{i+1}$  are not able to be connected in limited time, the returned pair of configurations has the nearest distance  $d_{min}$  between  $\mathcal{T}_i$  and  $\mathcal{T}_{i+1}$  which is defined as:

$$d_{min} = d(\mathcal{T}_i, \mathcal{T}_{i+1}) = \min\{d(q_a, q_b) | q_a \in \mathcal{T}_i, q_b \in \mathcal{T}_{i+1}\} \quad (2.2)$$

Where  $d$  is a metric on the configuration space, we use Euclidean Distance in our application.

Getting the knowledge of inaccessibility is usually not enough for the designer, because he may also want to know which surface or edge in the design limits the accessibility. The two disconnected configurations returned by local connection are used in our system to provide the interference information to the designer. Supposing that a virtual line is established by connecting the two returned configurations, the first contacting surfaces between the object and the obstacles when we move the object along the virtual line and from its initial point are the surfaces limiting the accessibility. However, changing the parameter related to these surfaces may not be necessary.

Therefore, the designer has the highest priority to decide to make the replanning motion or change the size of object.

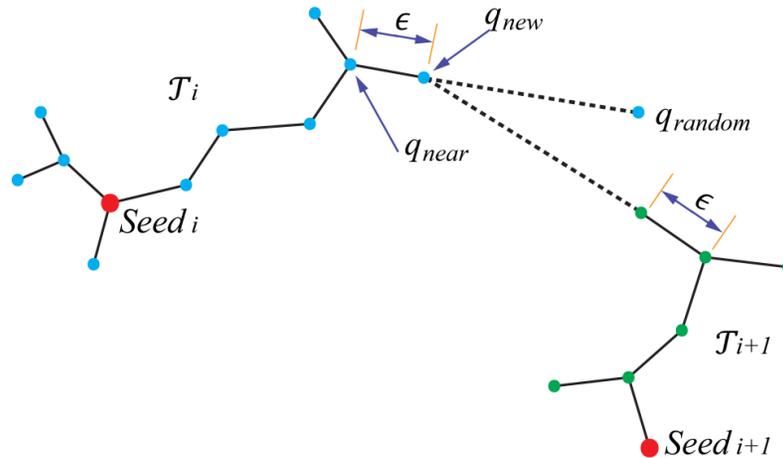


Figure 2.17. Connection of seeds by RRT-connect algorithm

### 2.4.3. Application cases

The following section describes how our interactive system works for the motion planning of a non-convex part passing through a narrow passage. We also demonstrate that the interaction with user equipped by a force feedback joystick in 2D and a space mouse in 3D can greatly decrease the search space.

#### 2.4.3.1. Application in 2D

All the experiments are performed in the software platform Matlab and on a PC Dual Core, 2.4 GHz and 2 GB ram.

In implementation, user manipulates the part in 2D environment with *Logitech Force 3D Pro* force feedback joystick. The moving object is allowed to have limited penetration area into the obstacles, and the maximal allowed penetration area is controlled by user, meanwhile any part of the moving object is forbidden to penetrate through the obstacle. And large penetration area is allowed in the interaction, but collisions only happen between the object and one obstacle at once. User can decide the beginning and end of the motion recording time.

Collision check is realized by detecting the intersected edges between two polygons. Penetration area is the area of overlap region between two polygons.

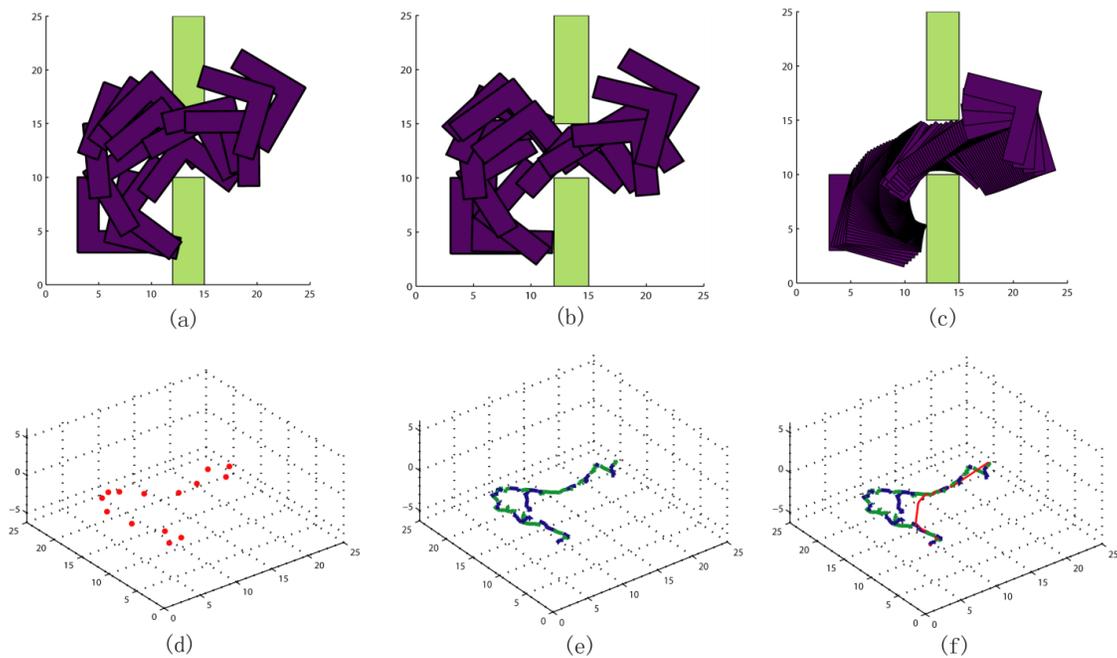


Figure 2.18. Successful case for an L shape part passing through a narrow passage

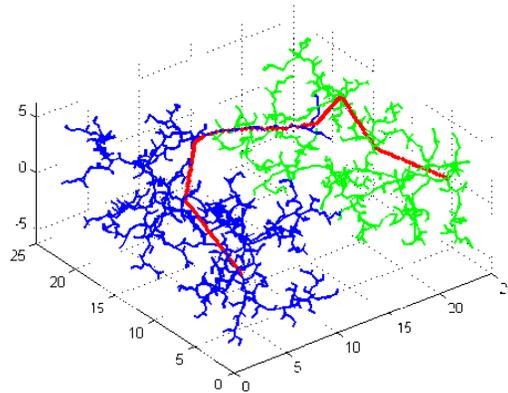


Figure 2.19. Exploration trace of basic BiRRT

In the situation of Fig. 2.18, user has to manipulate an L shape part to pass through a narrow hole of a wall in 2D environment. The moving part has three degrees of freedom. In this case, the hole is narrow enough that the planning with strict collision detection is time-consuming for both user and basic BiRRT planner. By relaxing the collision, user can quickly manipulate the part pass through the hole, as shown at Fig. 2.18 (a). This rough path manipulated by user is then retracted as shown at Fig. 2.18 (b) and the corresponding path in the configuration space is shown at Fig. 2.18 (d). As the number of configurations recorded for retraction is small, the retraction process doesn't require

much time. From every neighboring two configurations of retracted path, two trees (marked green and blue) are grown by BiRRT algorithm to connect each other, as shown at Fig. 2.18 (e).

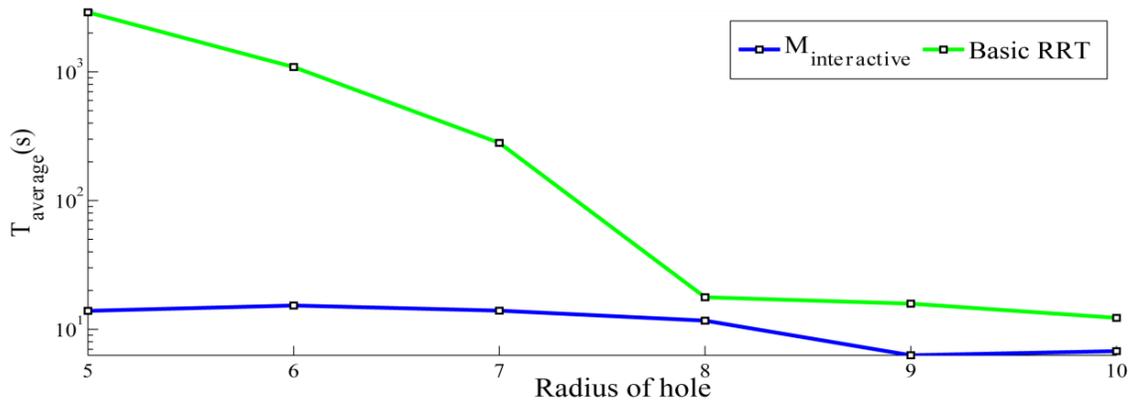


Figure 2.20. Comparison between basic BiRRT and our interactive process

Table 2.7. Comparison between BiRRT and our interactive process in case of radius = 5

	$T_{all}(s)$	$T_C(s)$	$T_R(s)$	$T_S(s)$	$Num_{node}$	$Num_R$
BiRRT	111.253	109.87	N/A	1.383	354	N/A
M-interactive	13.023	9.1900	1.239	2.594	93	5

$$T_C = T_{Connect}, T_R = T_{Retract}, T_S = T_{Smooth}, Num_R = T_{Retraction}$$

Table 2.8. Comparison between BiRRT and our interactive process in case of different radius

Hole Radius	Mean					
	5	6	7	8	9	10
BiRRT: $T(s)$	>2,901	>1,090	280.397	17.735	15.838	12.268
M-interactive: $T(s)$	13.91	15.314	13.965	11.676	6.291	6.771

A path successfully passing through the hole is found quickly by our interaction method shown at Fig. 2.18 (c) (f). In reason that our retraction process provides guidance to exploration of BiRRT, the number of visited configurations and tree search time of BiRRT algorithm are greatly reduced, as shown in Tab. 2.7. This result can be observed intuitively through comparison between the exploration trace of basic BiRRT algorithm in Fig. 2.19 and our interactive process in Fig. 2.18. From Tab. 2.7, we can also find how fast the random retraction process is, which only takes 1.239 seconds for 5 times retraction. We also make the comparison in cases of different radius of hole. For each radius of hole, we run both of the algorithms 5 times respectively, and take the average time. The results are demonstrated in Tab. 2.8 and Fig. 2.20. In challenging situation when the radius of hole is 5, the basic BiRRT will spend more than one hour to find a feasible solution. While in opposite, our method runs about 10 seconds. Moreover, in easy situations, our method also needs less time for planning. Through comparison, our interaction method appears more robust and efficient in planning motion of non-convex part in narrow passage.

In Fig. 2.21, the hole is designed too small for the part to pass through. However, intuitively detecting the infeasibility is difficult for users before the planning. In condition of relaxed collision, user can manipulate the part passing through the hole shown as Fig. 2.21 (b). After retraction process (shown as Fig. 2.21 (c) (d)) and connection process (shown as Fig. 2.21 (e)) of our method, the nearest two configurations respectively locating at entrance and exit of the narrow hole can be returned to user, shown as Fig. 2.21 (f). As a result, the designer can understand the size limitations of accessibility for a product design.

In Fig. 2.22, a Z shape part has to pass through cluttered environment from the lower left corner to the lower right corner. In this environment, many random pentagonal polygon obstacles are placed randomly. In the first three situations like in Fig. 2.21, these passages are actually neither possible for the part to pass through, nor easy for user to judge the connectivity in advance, due to the non-convex properties of part and obstacles.

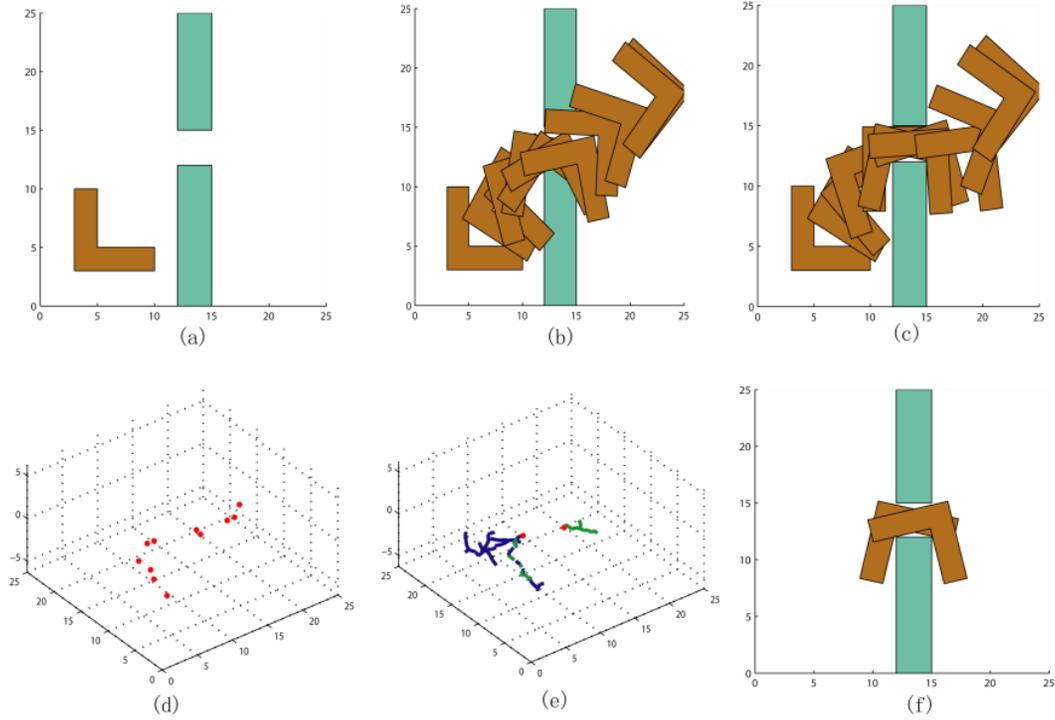


Figure 2.21. Failed case for an L shape part passing through a narrow passage

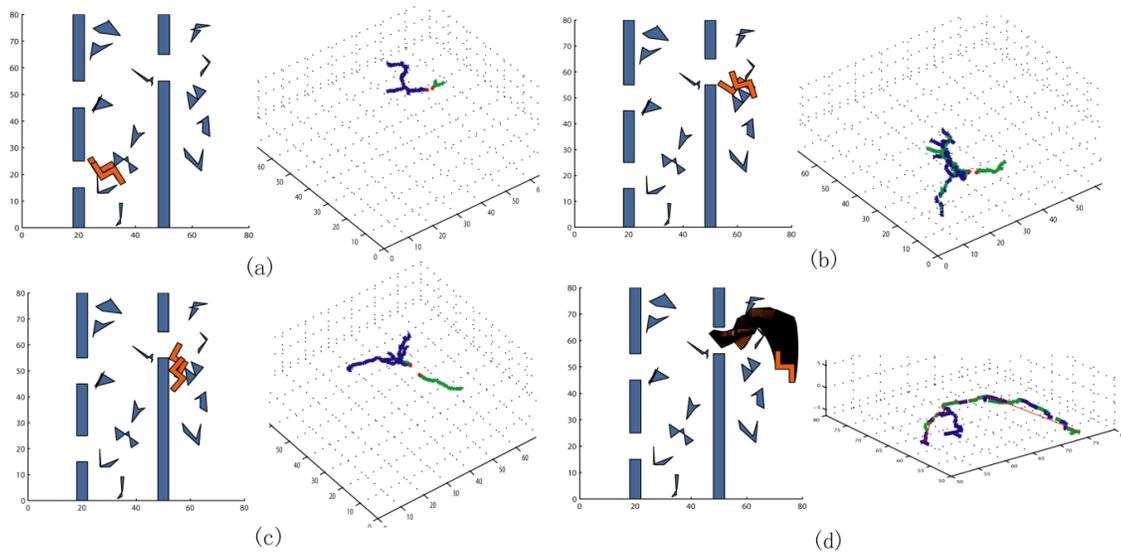


Figure 2.22. Simple case for a z shape part passing through narrow passages

However, after the interaction process of our method, the user can get the knowledge about the geometric constraints of every passage. These geometric constraints are the vertex, edge or surface of the obstacles which limits the connection of the nearest configuration pair. The geometric interference information can be easily obtained by detecting the contact between the obstacles and the nearest configuration pair. Finally, after another attempt, the user found a feasible solution in Fig. 2.22 (d). As a result, the designer can understand the size limitations to the accessibility for the whole product design. In this figure, we just displayed the planning results in difficult narrow passages, the intermediate process in relaxed passages were omitted.

#### **2.4.3.2. Application in 3D**

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

The CAD model is manipulated by users through the 3Dconnection's SpaceNavigator 3D mouse. The penetrating volume between the two CAD models is measured by the method previously introduced in the Section 2.3.3.

Our method is firstly applied to the flange benchmark shown in Fig. 2.23. The parameter  $N$  which controls the resolution of the sphere array is set as 25. In this benchmark, the opening of the obstacle is very narrow, inserting the pipe into the opening is usually very difficult when a user manipulates the object with the non-haptic devices and the collision is exactly checked. In contrast, our method relaxes the collision checking by allowing the penetration of object into the obstacles, thus the non-haptic 3D mouse can be used to comfortably manipulate the object. The step length of retraction is set as 0.05. The user manipulated path which consists of some in-collision configurations is shown in the first line of Fig. 2.23. Similar to the 2D cases, the different phases of the planning process are also demonstrated in Fig 2.23. In particular, three planning process in the different sampling rates are also demonstrated. The sampling rate controls the number of configurations picked up from the user manipulated path. Our method and the RRT-connect algorithm are applied to the benchmark in the conditions that the flange is in the different scale of its original size. The larger the scale is, the narrower the passage in the configuration space become. For each scale, the experiment has been run five times and the average data is taken. The experimental results are shown in the Tab. 2.9 ~ 2.12. Compared to the RRT-connect algorithm, our method can reduce the search trials, collision-checks and planning time by an order of magnitude when the scenario is tightly constrained. Meanwhile, in the easy situations, our method did not lose out to the RRT-connect algorithm. It is also noticed that, in our method, the larger sampling rate can increase the chances of making the retractions. When the scenario is tightly constrained, since the retraction step is much faster than the connection step, raising the sampling rate is the better choice.

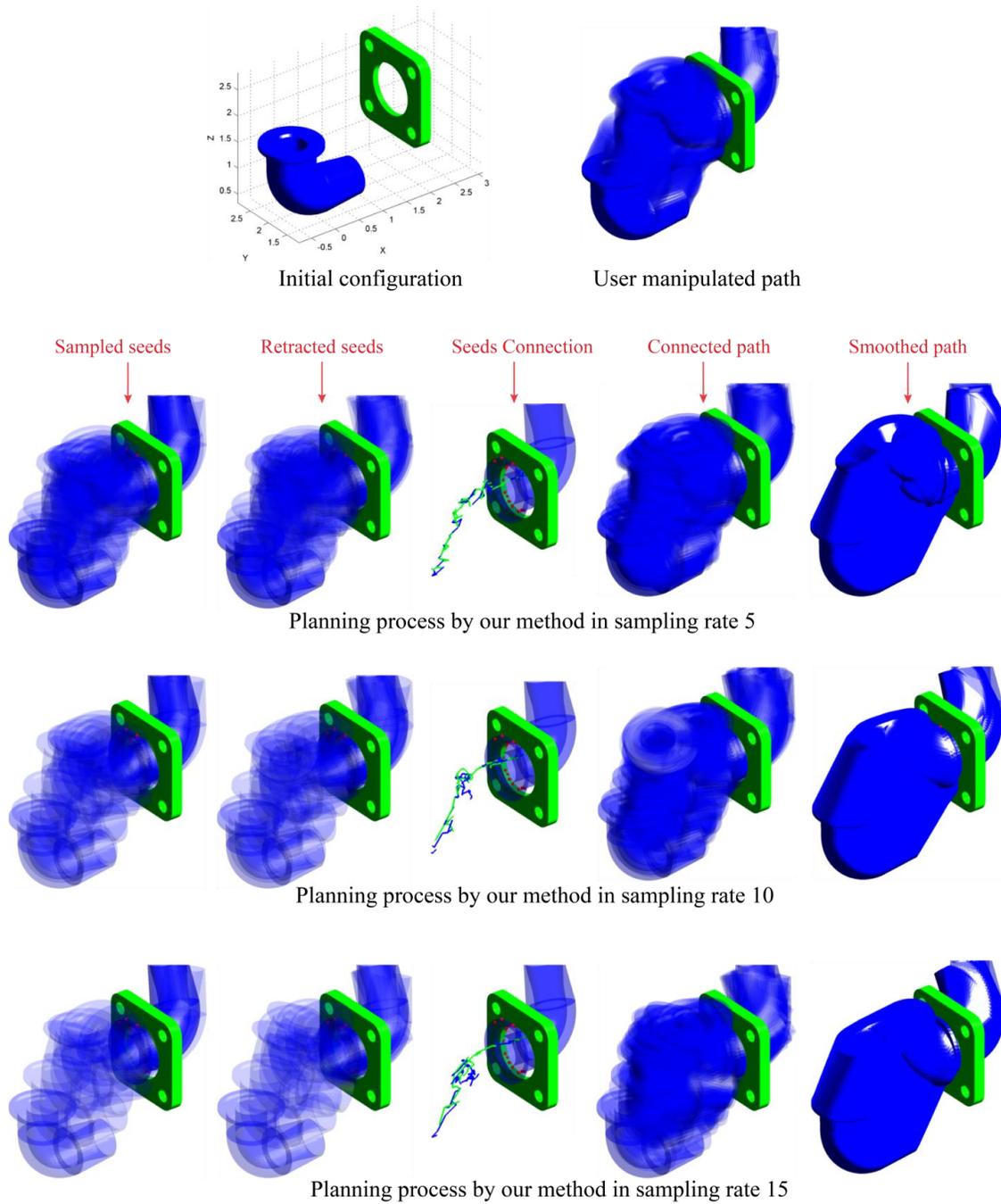


Figure 2.23. The Planning process by our method in different sampling rates

Table 2.9. Planning result by RRT-connect

Scale	Connection Time	Node Searched	Collision Check Calls
0.90	172.3	3595.2	3595.2
0.85	104.9	2260.0	2260.0
0.80	34.1	745.2	745.2
0.75	25.4	543.6	543.6
0.70	22.4	480.8	480.8

Table 2.10. Planning result by our method in the sampling rate 5

Scale	Total Time	Retract Time	Connect Time	Retract Node	Node Searched	Collision Check Calls
0.90	23.4	11.2	12.3	8.2	325.2	698.0
0.85	26.4	10.1	16.2	8.6	416.8	754.8
0.80	16.7	5.8	10.9	7.0	277.6	494.0
0.75	15.1	6.2	8.9	7.2	242.0	476.2
0.70	14.2	4.8	9.4	5.4	241.2	444.4

Table 2.11. Planning result by our method in the sampling rate 10

Scale	Total Time	Retract Time	Connect Time	Retract Node	Node Searched	Collision Check Calls
0.90	24.2	3.8	20.5	4.0	592.4	775.4
0.85	19.9	3.6	16.3	4.2	424.0	598.2
0.80	22.9	3.0	19.8	3.8	498.4	643.4
0.75	15.2	3.1	12.9	4.0	324.0	477.2
0.70	11.9	1.9	10.0	2.4	257.6	386.4

Table 2.12. Planning result by our method in the sampling rate 15

Scale	Total Time	Retract Time	Connect Time	Retract Node	Node Searched	Collision Check Calls
0.90	47.9	3.1	44.7	2.8	1181.2	1348.4
0.85	22.7	2.4	20.3	2.8	525.6	668.6
0.80	13.9	2.5	11.4	2.4	290.8	422.2
0.75	12.5	2.1	10.4	1.8	272.4	400.0
0.70	11.3	1.5	9.8	2.0	237.6	355.8

## 2.5. Conclusion

In this chapter, an interactive motion planning system was proposed for the accessibility verification mission in product design. In considering that human has the ability to find a motion plan in loose environment, we allow the manipulated object to have some penetration into the obstacle, which helps to relieve the user from the burdensome manipulation in complex environment. Depending on an efficient and robust retraction method, the in-collision part of user's manipulation trajectory can be retracted to some seeds in the configuration space. In feasible condition, these seeds can be connected by RRT-connect algorithm in pairs to find a valid motion trajectory from the initial configuration to the goal configuration. Otherwise, the interference information is provided to the designer for improving the design parameters. Even though the connection phase by automatic motion planners is carried out after the human's manipulation in our system, the retraction and connection strategy is compatible with other interactive motion planning methods which allow the simultaneous cooperation between the user and the automatic planners due to the efficiency of the

retraction process. The retracted configurations from user's manipulation trajectory can be added to the expanding tree of automatic motion planners when the retraction is in real time. The added retracted configurations can greatly accelerate the exploration of the sampling based algorithms in narrow passages, which has been proved by the experimental results of our method.

However, an essential problem always bothering us is what is the real reason limiting the improvement of interactive motion planning method. The answer to the question is not limited to the difficulties in virtual world introduced by the interactive technique, because many complex tasks are also difficult for humans to deal with in the real life. Before continuing to answer the question, we find an interesting phenomenon that human can improve their manipulation skill after a long-term practical training. This capability benefits from the learning mechanism of a human's nervous system, which is not explicit to date. In the next chapter, we will introduce another work which helps to learn from the motion experience in a novel perspective.

# 3 Motion Learning Based on Scenario Matching

---

3.1. Introduction.....	59
3.2. The hypothesis from an observation.....	60
3.3. Overview of the motion learning process.....	64
3.4. Scenario description and skeleton generation.....	65
3.4.1. Medial axis.....	65
3.4.2. Curve skeleton.....	65
3.4.3. Scenario skeleton generation.....	66
3.4.4. Scenario segmentation.....	68
3.5. Similarity measurement of scenarios by DTW.....	69
3.5.1. Feature selection on scenario skeleton.....	70
3.5.2. Similarity measurement of scenarios by Dynamic Time Warping (DTW).....	71
3.5.3. Conclusion.....	77
3.6. Organization of scenario segments in motion library.....	77
3.6.1. Indexing structure of scenario segments.....	78
3.6.2. R-tree introduction.....	80
3.6.3. Clustering scenario segments with R-tree.....	82
3.7. Conclusion.....	87

---

## 3.1. Introduction

Usually two valuable phenomena exist when human manipulates the object in an accessibility verification task. The first phenomenon is about the influence of narrow passage, meaning that the difficulties of manual manipulation increase when the environment becomes tightly constrained. Meanwhile, the narrow passage also ties the foot of the automatic motion planner. In the previous

chapter, the collision checking is relaxed and a novel retraction method is integrated with the automatic motion planner to reduce the impact of narrow passage. However, the improvement in some complex situations is limited, because the accurate manual manipulation is still inevitable. Another phenomenon which may help to improve the efficiency of interactive motion planning is that, humans can learn the experience from the past practice. However, this kind of knowledge is not explicit and hard to be represented at present. The concept of online motion learning based on the topological similarity between scenarios has been proposed in (Yan, Poirson et al. 2013), however, no concrete or workable method was developed.

In this chapter, a novel motion learning mechanism is proposed in (Yan, Poirson et al. 2015) based on a hypothesis, which explores the relationship between the topological structure of scenario and the movement of object. Motion planning is regarded as constructing the mapping from the workspace to the configuration space according to the similarity of the scenarios.

## 3.2. The hypothesis from an observation

The motion planning problem is always regarded as finding a path in the collision-free configuration space  $\mathcal{CS}_{free}$  which can connect the initial and goal configurations. Following this perspective, most of the methods proposed in the motion planning field aim to answer the question about how to move the object in avoidance of collision. The approximate simulation of configuration space  $\mathcal{CS}$  and the computation based on the information of  $\mathcal{CS}_{obstacle}$  are always the central concerns. However, few works in the literature try to answer the question about how the topological structure of space enclosed by the obstacles influences the topological structure of the  $\mathcal{CS}_{free}$ . Because the computing of  $\mathcal{CS}_{free}$  is complex and time-consuming at present, this kind of relationship between the topology of the scenario and the topology of  $\mathcal{CS}_{free}$  is not explicit. In this chapter, we try to provide a novel perspective to explore the relationship by measuring the similarity between scenarios.

Some concepts are necessary to be clearly defined before the further discussion. The workspace we mentioned here is the subset of the three-dimensional Euclidean space which contains the moving object and the obstacle, and can be noted as  $\mathcal{WS} \subset \mathcal{R}^3$ . The subset of  $\mathcal{WS}$  taken up by the obstacles is noted by  $\mathcal{WS}_{obstacle}$ . The scenario for a moving object is the subset of the  $\mathcal{WS}$  which does not contain the obstacle and is noted as  $\mathcal{WS}_{scenario}$ ,  $\mathcal{WS}_{scenario} = \{x | x \in \mathcal{WS}, x \notin \mathcal{WS}_{obstacle}\}$ .

Through observations, we have found that there exists some kind of certainty for the motion of an object when its scenario is tightly constrained. This can be understood through an example which is shown in Fig. 3.1. When a smooth motion trajectory of the object is obtained like in Fig. 3.1 (a), a

sequence of spheres can be constructed, each one of which touches at least two surfaces of the surrounding obstacles in Fig. 3.1 (b). These spheres indicate the maximal volume of workspace in which the object can move. The red curve is the centerline of these spheres and its radius change is shown beneath itself. The segmentation of motion trajectory in the narrow hole and its scenario segment are selected and shown in Fig. 3.1 (c). Even though making the object to pass through the narrow hole is difficult for both human and automatic algorithm, the way of moving the object that contains the simultaneous slight translations and rotations is the only way with which the object can successfully pass through the hole.

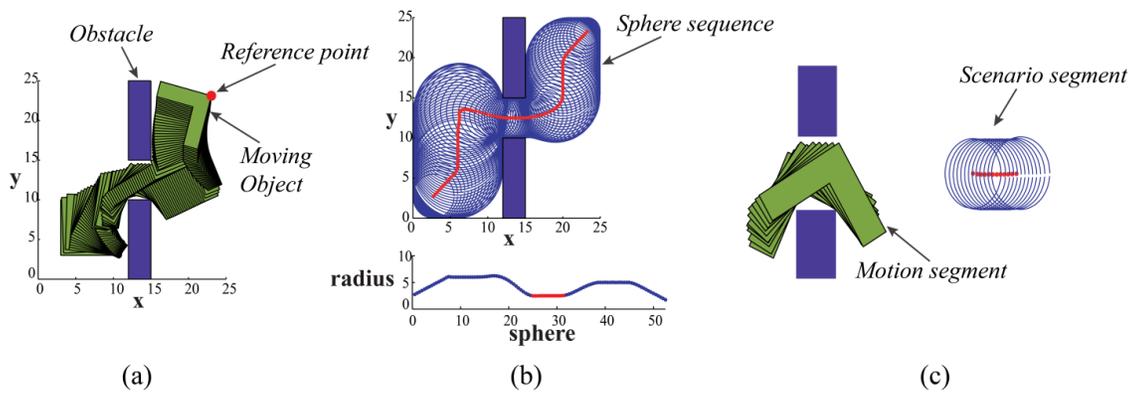


Figure 3.1. Motion segment learned in a successful planning task

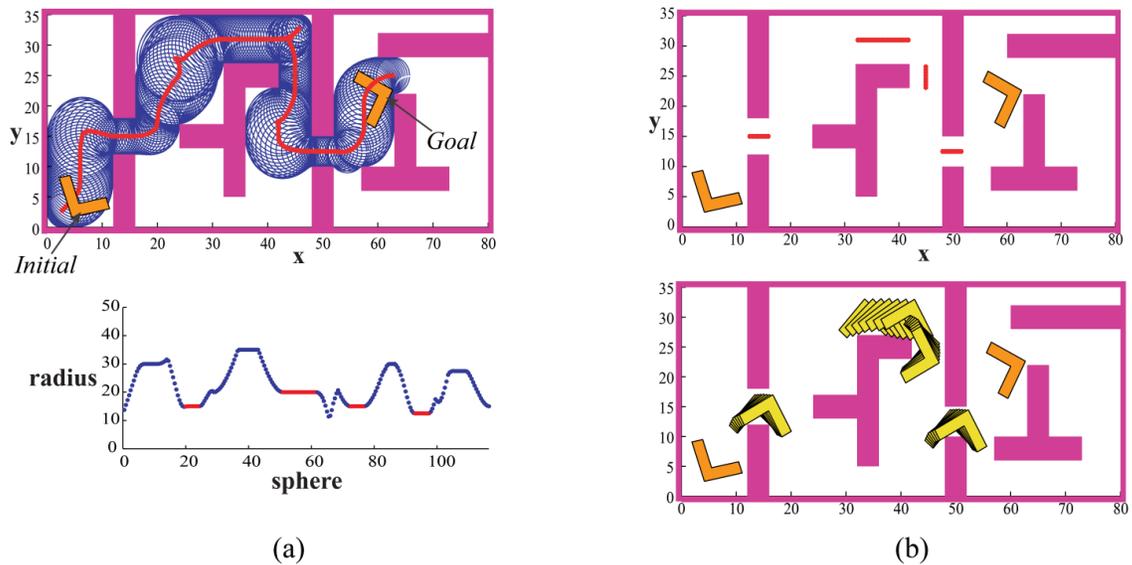


Figure 3.2. Learned motion segment is applied to the four similar scenarios

Since the certainty exist between the motion trajectory and its scenario when the scenario is tightly constrained by obstacles, the movements of object should obey the same motion pattern in two scenarios which are similar to each other. Fig. 3.2 shows an example in which the motion segment learned in the previous example can be directly applied to the similar scenarios. As the previous case, the sphere sequence is constructed to indicate the maximal volume in which the object can move, shown in Fig. 3.2 (a) up. From the radius change of these spheres in Fig. 3.2 (a) bottom, it is noticed that there exist four difficult passages in the new environment. The centerlines of the spheres in these four passages are shown in Fig. 3.2 (b) up. Intuitively, the four passages are similar to the scenario segment selected in the previous example. Based on their similarity, the learned motion segment can be directly applied to the four passages as shown in Fig. 3.2 (b) bottom. The connection between these new motion segments is much easier than finding them. The above mentioned intuitive experience implies a novel perspective for solving the narrow passage problem.

For the further analysis of this intuitive experience, the computation of the complete  $\mathcal{CS}_{free}$  is necessary, which has been studied in (Varadhan, Kim et al. 2006, Nelaturi and Shapiro 2013) by the adaptive sampling. However, computing the complete  $\mathcal{CS}_{free}$  cannot fulfill the efficiency requirement. Hence the sampling based motion planners usually search out a collision-free motion path on a graph which is established by randomly taking samples in  $\mathcal{CS}$  and connecting the adjacent samples with collision-free edges. The performance of the sampling strategy degrades in the narrow passages of  $\mathcal{CS}_{free}$ . An ideal solution to this problem is to find a low-cost function  $f$  which can approximately compute  $\mathcal{CS}_{free}$  from the workspace  $\mathcal{WS}$ ,  $f: \mathcal{WS} \rightarrow \mathcal{CS}_{free}$ . Unfortunately, the computational complexity of  $f$  is not lower than that of  $\mathcal{CS}_{free}$ . In another perspective, for a given object, a mapping  $f_1$  from the current scenario  $\mathcal{WS}_{scenario}^1$  to the Boolean value of accessibility can be found:

$$f_1 : \mathcal{WS}_{scenario}^1 \rightarrow b, b \in \{accessible, inaccessible\} \quad (3.1)$$

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

$$f_2 : \mathcal{WS}_{scenario}^1 \rightarrow p, p \subset \mathcal{CS}_{free} \quad (3.2)$$

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

$$D : \mathcal{WS}_{scenario} \times \mathcal{WS}_{scenario} \rightarrow d, d \in R \quad (3.3)$$

and  $\mathcal{D}(\mathcal{WS}_{scenario}^1, \mathcal{WS}_{scenario}^2) < \varepsilon$  where  $\varepsilon$  is a threshold, is it possible to apply the path  $p$  to the new scenario  $\mathcal{WS}_{scenario}^2$ ? Or if the mapping  $f_2$  fits a set of scenarios which are similar to  $\mathcal{WS}_{scenario}^1$ . In Fig. 3.3, 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. 3.3, the narrowest part of the three scenarios are 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 to the centerline of the maximal sphere sequence. The C-contact space  $\mathcal{CS}_{contact}$  of the three scenarios is computed through uniformly sampling the configuration space, and is shown by the blue surface in Fig. 3.3. It is obvious that the topological structure of  $\mathcal{CS}_{free}$  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 learning framework.

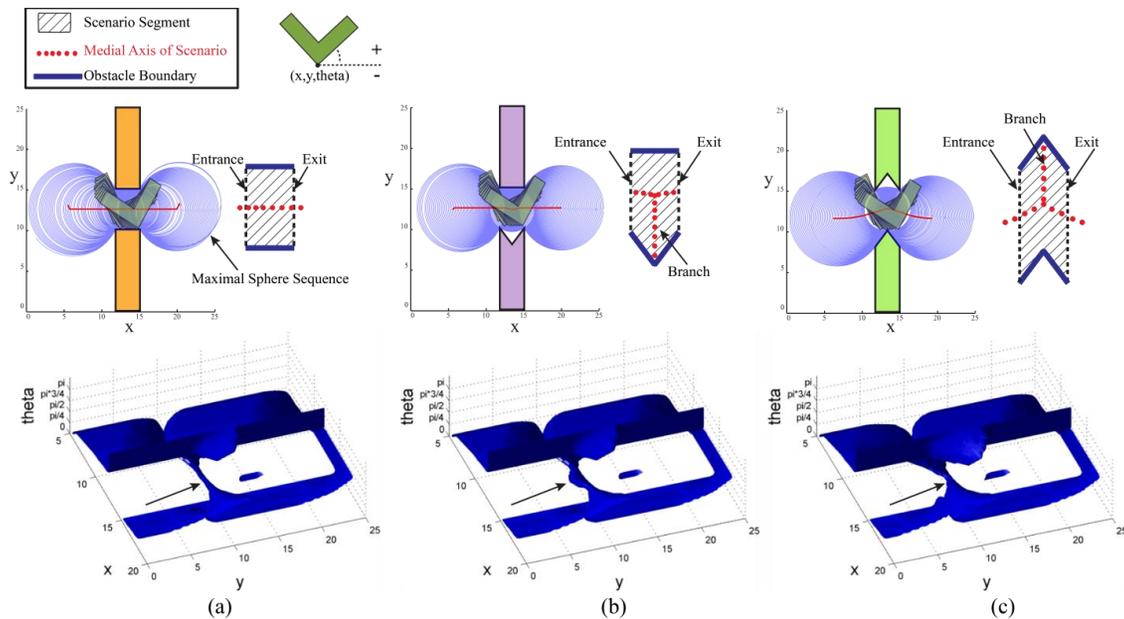


Figure 3.3. Similar scenario segments and their topological representations

### 3.3. Overview of the motion learning process

In the last chapter, the interactive motion planning method is implemented to verify the accessibility of a product design, which can be regarded as planning the motion from scratch because every new planning task is treated in the same way without the support of any experience. However, the situations about human's behavior run in an opposite way. The skills of manipulation mastered by the assembly experts always become better and better after the constant practice. This kind of expertise is still difficult to describe and model at present, due to the complexity of the nervous system. Besides, the participation of users without expertise is more and more strongly required in the early phase of product design. Therefore, we construct a novel motion learning method which can extract the motion experience from the interactive motion planning process and can help to reduce the difficulty of manipulation and accelerate the efficiency of automatic motion planner.

In our motion learning process, the sphere sequence which has been described in the example of the previous section is generated when a collision-free motion trajectory is found or the conclusion of non-accessibility is verified in the interactive motion planning process. Then the sphere sequence regarded as the motion scenario splits into segments for efficiency and parallel processing reason. The motion trajectory also splits into segments according to the segmentation of scenario. After segmentation, several pieces of motion segment and their scenario segment are saved into motion library, in which the scenario segments are organized by a hierarchical structure according to the similarity. The overview of the learning process is shown in Fig. 3.4.

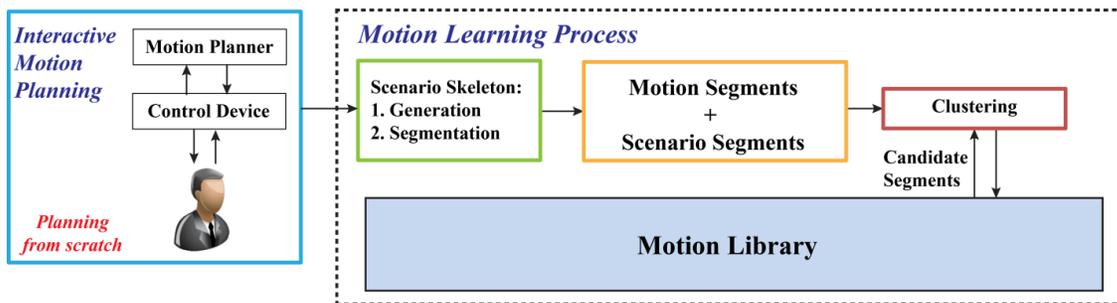


Figure 3.4. Overview of the motion learning process

Some salient characteristics of our motion learning method need to be noticed before the detailed explanation of every functional module:

- 1) Our motion learning process is compatible with any kind of interactive or automatic motion planning method.

- 2) The learning mechanism is established based on the hypothesis stated in the previous section. The similarity of scenario segments is measured by the topological representation and volume change of the scenario. The learned motion experience is highly adaptive to be applied into new environment.
- 3) The hypothesis is more likely to be valid when the scenario is narrower. This is the reason that our method can help to solve the “narrow passage” problem which is the bottleneck of the sampling based algorithms.
- 4) Our method can constantly evolve from an empty motion library and maintenance the motion library in the whole life of the object.
- 5) The retrieval of motion segment is highly efficient due to the low computational complexity of indexing method and the hierarchical organization structure in the motion library. A promising aspect is that it is feasible to retrieve the motion in real time from the motion library with millions of motion segments.

In the following context, three sections are dedicated to explain the method of generating and splitting the scenario, the method of measuring the similarity of scenarios and the method of organizing scenario segments in the motion library. The salient features will also be demonstrated with the explanation and the experimental results.

## **3.4. Scenario description and scenario skeleton generation**

In this section, the motion scenario has been described by the local maximal sphere sequence. The centerline and the radius change of the sphere sequence can respectively describe the topological structure and the volume change of the scenario. The topological structure of scenario is named as the “scenario skeleton”, which is defined as the medial axis and curve skeleton with branches pruned.

### **3.4.1. Medial axis**

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 (Ma, Bae et al. 2012). 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 the translational exploration in the 2D plane (Geraerts and Overmars 2007). In 3D application, the sampled configurations are retracted onto the medial axis of the collision-free configuration space (Lien, Keyser et al. 2006, Yeh, Denny et al. 2013). However, this method also endures some expensive computations like other retraction based motion planners.

### 3.4.2. Curve skeleton

Curve skeletons are 1D curves which are locally centered in the shape (Kustra, Jalba et al. 2013). The curve-skeleton captures the essential topology of the underlying object in an easy to understand and very compact form. Examples of applications that use a curve skeleton include: virtual navigation, registration, animation, morphing, scientific analysis, shape recognition, and shape retrieval (Cornea, Silver et al. 2007). Based on the different representations of object, there are a great number of curve skeleton extraction algorithms. For the proper classification and briefly overview of these algorithms, readers can refer to (Jiang, Xu et al. 2013). In the shape recognition field, the curve skeleton is usually applied to describe the articulate systems of objects (Kin - Chung Au, Tai et al. 2010, Chang and Kimia 2011), and the calculation is time-consuming which is far beyond the requirements of interaction. Since the topological structure of a scenario is the medial axis and curve skeleton with branches pruned, the scenario recognition is actually a curve matching problem.

### 3.4.3. Scenario skeleton generation

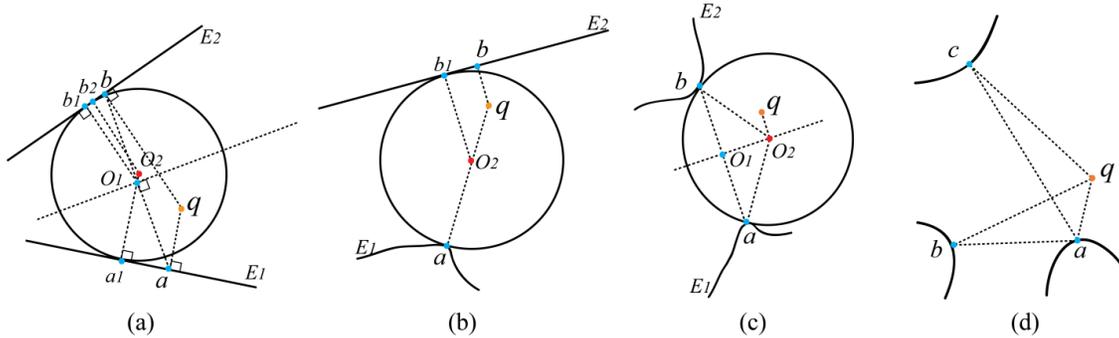


Figure 3.5. Sphere generation according to the position of the reference point

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. 3.5, 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 as the equal function.

- 1) *Surface-surface*: in Fig. 3.5 (a), the collision detection algorithm finds out two nearest surfaces  $\{E1, E2\}$  to the reference point  $q$ . The points  $\{a, b\}$  on the surfaces  $E1$  and  $E2$  are the perpendicular foets from the point  $q$ . From the center point  $O1$  between  $a$  and  $b$ , two perpendicular foets  $\{a1, b1\}$  on the surfaces  $E1$  and  $E2$  can be also found. Through simple vector operations, a local sphere with center  $O2$  which is tangent to the surfaces  $E1$  and  $E2$  on  $a1$  and  $b2$  is created.
- 2) *Surface-vertex*: in Fig. 3.5 (b), the surface  $E2$  and the vertex  $a$  on the surface  $E1$  are the nearest to the reference point  $q$ . The point  $b$  on the surface  $E2$  is the perpendicular foot from the point  $q$ . Through simple vector operations, a local sphere with center  $O1$  which is tangent to the surfaces  $E1$  and  $E2$  on  $a$  and  $b1$  is created.
- 3) *Vertex-vertex*: in Fig. 3.5 (c), two vertices  $\{a, b\}$  on the surface  $E1$  and  $E2$  are the nearest to the reference point  $q$ . On the line which is vertical to the line  $ab$  and passes the center point  $O1$  of the line  $ab$ , the center of sphere  $O2$  which is the perpendicular foot from the point  $q$  is found. The sphere is tangent to the surfaces  $E1$  and  $E2$  on  $a$  and  $b$ .

In the situation shown in Fig. 3.5 (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  $b$ .

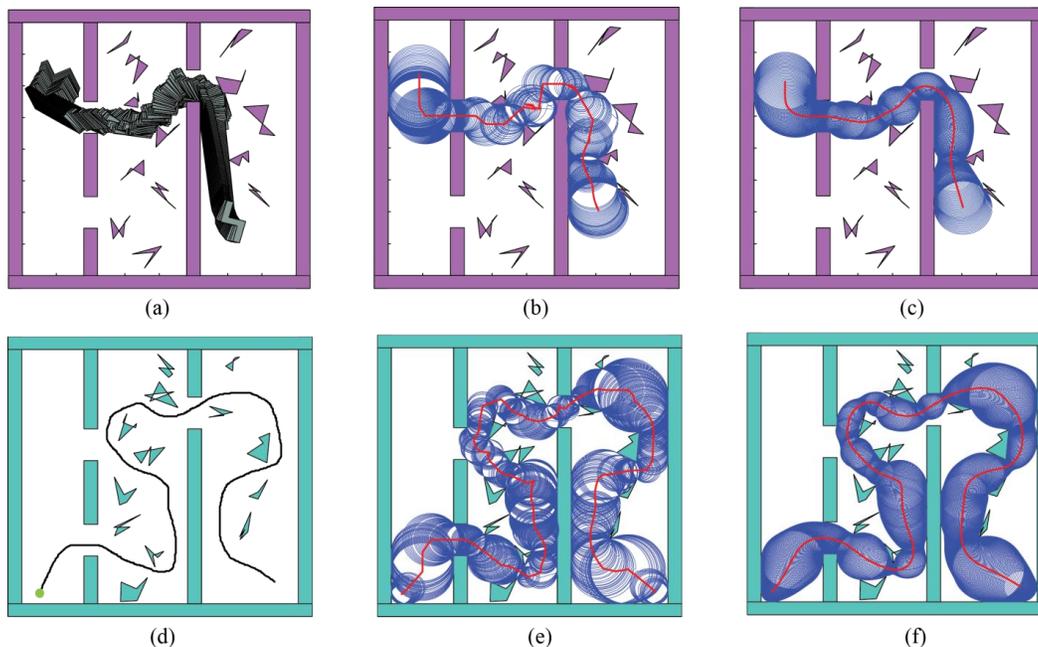


Figure 3.6. Sphere generation in a planar cluttered environment in two cases

Fig. 3.6 shows the generation of the scenario skeleton in two cases. The first case happens after a motion path is found in Fig. 3.6 (a), in which the spheres are generated according to the trace of a vertex of the object in Fig. 3.6 (b). In the second case, according to the trace of the mouse's cursor which is manipulated freely by a user shown in Fig. 3.6 (d), a sphere sequence is generated in Fig. 3.6 (e). All the centers of spheres constitute the skeleton of a scenario. The jitters on the skeleton may damage the scenario matching which will be introduced in the following sections, thus the smoothing is essential, shown in Fig. 3.6 (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.

#### 3.4.4. 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. Each scenario segment can be seen as a *scenario primitive* and its contained motion can be seen as a *motion primitive*.

The segmentation is based on the changing trend of spheres' radius. The narrow passages usually appear with phenomenon in which radius of spheres shrink firstly and then re-enlarge. The beginning and ending positions for this kind of wave on radius curve can be detected by the curvature of 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

$$k = \frac{u'v'' - v'u''}{(u'^2 + v'^2)^{3/2}} \quad (3.4)$$

We adopt the method in (Kroon 2011) to calculate the curvature, which firstly fits a quadratic polynomial to the local points and then find out the derivatives of  $u$  and  $v$ . In the rest of this thesis, the curvature calculation is the same.

In Fig 3.7 (a), the segmentation of a scenario is demonstrated. The radius change of the spheres is shown in Fig 3.7 (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  $\varepsilon$  set by users, if the bottom of a valley is lower than  $\varepsilon$ , 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 3.7 (a).

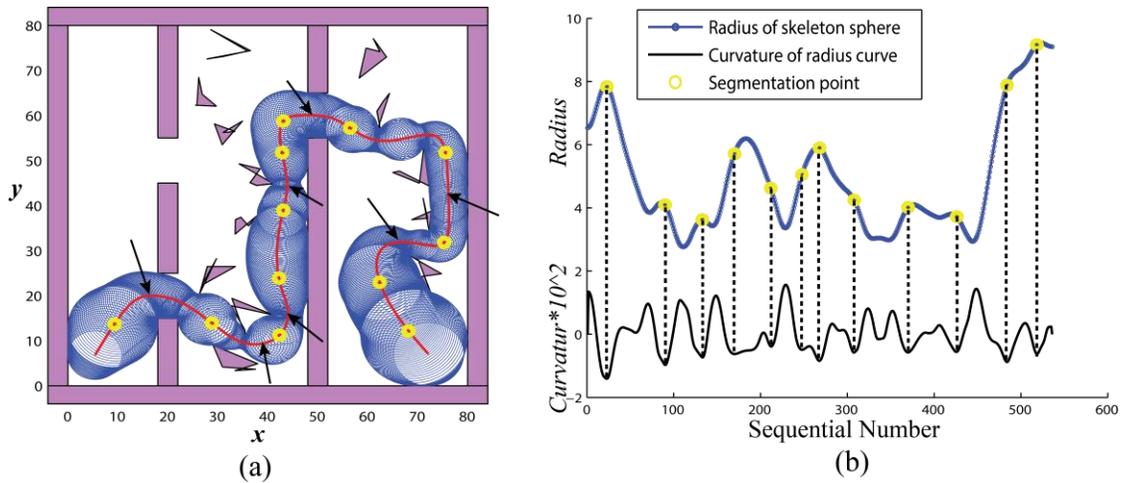


Figure 3.7. Segmentation of scenario

### 3.5. Similarity measurement of scenarios by DTW

After segmentation, the scenario is split into several pieces of segments. For every piece of scenario segment, there exists a segment of motion trajectory corresponding to it. In the section 3.2, a hypothesis related to the object's movement is assumed which indicates that the motion of the object is similar in two scenarios which have similar topological structure and volume change. Therefore, supposing an unknown scenario segment is obtained in a new environment, if we could find a similar scenario segment from the experience, the motion in the unknown scenario segment should obey the similar way of moving in the learned scenario segment, which is the foundation of our motion learning mechanism. Since the scenario is represented by the sphere sequence, the similarity of scenario is measured by comparing the sphere sequence of two scenarios. Therefore, the topological structure of the scenario is the centerline (skeleton) of sphere sequence and the volume change is represented by the radius change. Since both of the centerline and radius change of scenario are 1D curve, the similarity measurement of scenario is actually the curve matching problem.

Shape of an object, represented by its contour, is one of the most important visual features that are thought to be used by humans to determine the similarity of objects (Khalid 2012). Therefore, curve matching is one of central issue in computer vision, which has wide applications in time series analysis (Agrawal, Faloutsos et al. 1993), shape matching (Gope, Kehtarnavaz et al. 2005), speech

recognition (Müller 2007), signature verification (Cui, Femiani et al. 2009), human video retrieval (Nguyen, Merienne et al. 2010) and similarity search in trajectory database (Pelekis, Kopanakis et al. 2007). In data mining field, curve matching has attracted more and more attention in discovering the pattern of curves and retrieving the most similar curve in curve database.

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. In (Agrawal, Faloutsos et al. 1993), time series curve have been represented into coefficients in the results of Discrete Fourier Transformation (DFT), beyond which distance between two curves could be calculated. The coefficients seen as multidimensional coordinates provide possibility for hierarchical clustering and multidimensional indexing like R-tree. However, it's not easy to find the correspondence with this method. In our case, we also need to find out the correspondence between two scenarios for motion transformation. Another category of algorithms use dynamic programming to find the correspondence by the shortest path on an accumulated distance matrix and measure the similarity by the length of this path. These algorithms include Dynamic Time Warping (DTW) (Berndt and Clifford 1996), Longest Common Sub Sequence (LCSS) (Bollobás, Das et al. 1997) and Edit Distance on Real sequences (EDR) (Chen, zsu et al. 2005). An exact indexing method of dynamic time warping has been demonstrated in (Keogh and Ratanamahatana 2005), in which a piecewise aggregate lower bounding technique was used to model a sequence with a linear combination of box basis functions. Due to its intensive study and wide application, we will use dynamic time warping as our similarity metric in this paper.

### 3.5.1. Feature selection on scenario skeleton

Since the topological structure of scenario is represented by its skeleton (centerline of sphere sequence), the movement of object has high correlation to the changing trend of the scenario skeleton. Specifically, the object usually rotates and translates by a larger quantity when the corresponding scenario skeleton appears with a greater degree of bending. Therefore, the feature selected on scenario skeleton should be able to describe this change of bending. Moreover, the selected feature of skeleton curve should meet Euclidean Transformation, which means that it is invariant to rotation and translation, because the location and orientation of the scenario segments has no impact on the movement pattern of object. Under the transformation of scaling and shearing, the correspondence between movement of object and scenario skeleton is still not clear at present time. The hypothesis about motion assumed in previous section is more like a description to the relationship between the movement pattern of object and the geometrical feature of local space enclosed by obstacles. It seems senseless if we compare a short scenario segment to a much longer scenario segment, because a segment of motion trajectory in the local environment doesn't have obvious relationship with a complete motion trajectory in the whole task. The scenario skeleton in

other affine transformation such as shearing will change the bending degree, in which situation it is still not explicit how to make motion transformation.

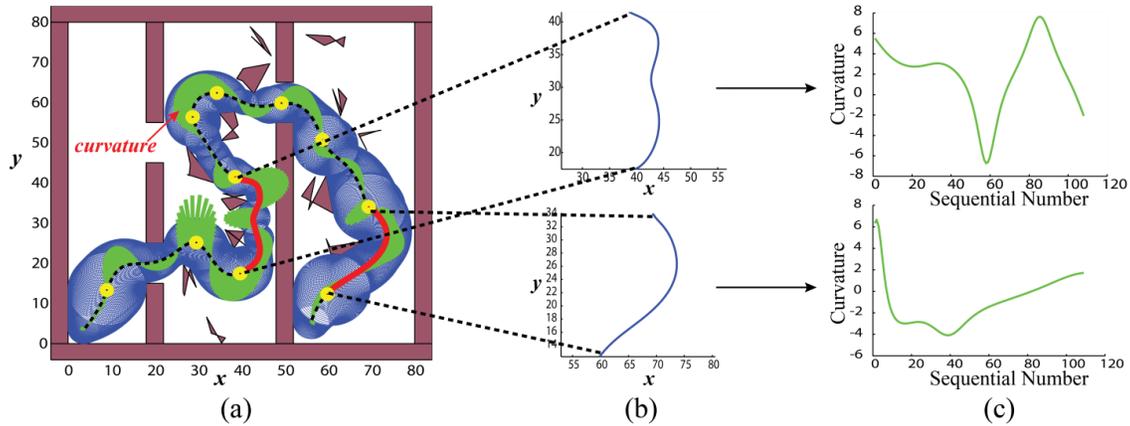


Figure 3.8. Calculation of scenario skeleton's curvature

Therefore, we adopt the curvature as feature to describe the bending degree of the scenario skeleton, which is rotational and translational invariant. In Fig. 3.8, we demonstrate the curvature change on two scenario skeleton segments. In Fig. 3.8 (a), the skeleton of the scenario is marked by the dashed curve. Then we zoom in on two segments of the scenario skeleton which is marked by the red solid curve and show them in Fig. 3.8 (b). The curvature changes of these two skeleton segments are shown in Fig. 3.8 (c).

### 3.5.2. Similarity measurement of scenarios by Dynamic Time Warping (DTW)

After previous operations, the similarity measurement becomes comparing the similarity of sphere centerline's curvature change and sphere sequence's radius change of two scenarios. The curvature and radius change are two curves in their own space, which are sampled into two sequences. Therefore, an appropriate distance metric is required to measure the similarity of curvature sequence and radius sequence between two scenarios. Besides, the distance metric is also required to provide the accurate alignment between two sequences, because the result of similarity measurement will serve motion transformation in the motion generation phase which will be talked about in the next chapter. The motion segment can be applied into new scenario segment only after every configuration in this motion segment is transformed according to the correspondence between the learned scenario segment and the new scenario segment. Dynamic time warping (DTW) is a well-known technique to find an optimal alignment between two given sequences under certain restrictions. Intuitively, the sequences are warped in a nonlinear fashion to match each other. DTW

has been applied to temporal sequences of video, audio and graphics data - indeed, any data which can be turned into a linear sequence can be analyzed with DTW.

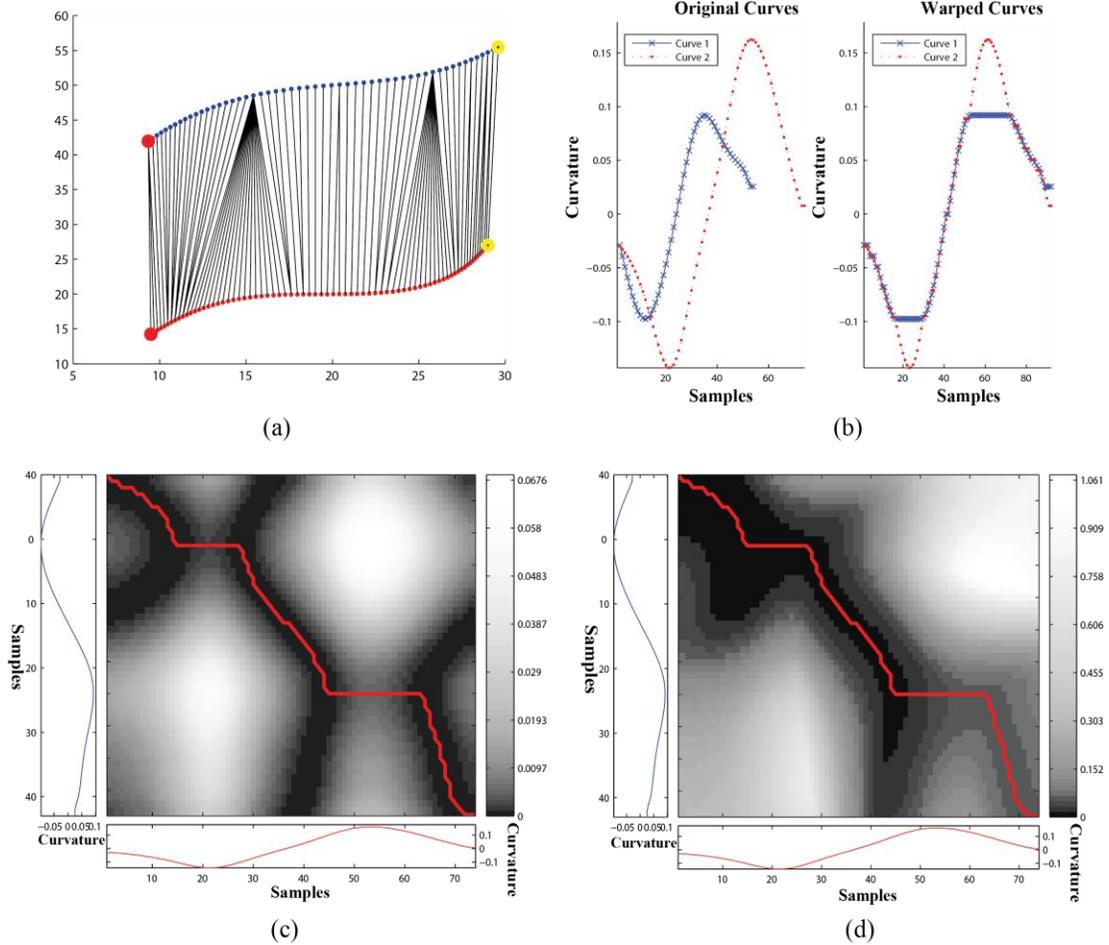


Figure 3.9. Alignment between two similar skeleton segments based on curvature.

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  $M \in \mathbb{N}$  in space  $\mathcal{F}$ . 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. Therefore, the value in  $\mathcal{F}$  can be curvature or radius in our application. The  $x_n, y_m \in \mathcal{F}$  for  $n \in [1:N]$  and  $m \in [1:M]$ . For the skeleton of two scenario segment shown in Fig. 3.9 (a), the change of their curvature before and after the alignment by DTW is shown in Fig. 3.9 (b), in which the  $x_n, y_m$  are respectively the curvature values of the two skeletons.

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  $C(n, m) := c(x_n, y_m)$ , as shown in Fig. 3.9 (c).

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. We will calculate the dynamic time warping by the same definition and formulation in (Müller 2007).

A  $(N, M)$  warping path is a sequence  $p := (p_1, p_2, \dots, p_L)$  with  $p_l = (n_l, m_l) \in [1:N] \times [1:M]$  for  $l \in [1:L]$  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  $l \in [1:L - 1]$ .

A  $(N, M)$  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 boundary condition enforces that the first elements of  $X$  and  $Y$  as well as the last elements of  $X$  and  $Y$  are aligned to each other. The step size condition ensures that all index pairs contained in a warping path  $p$  are pair wise distinct.

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:

$$c_p(X, Y) = \sum_{l=1}^L c(x_{n_l}, y_{m_l}) \quad (3.5)$$

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

$$DTW(X, Y) = c_{p^*}(X, Y) = \min\{c_p(X, Y) \mid p \text{ is an } (N, M) \text{ warping path}\} \quad (3.6)$$

The optimal warping path can be found by dynamic programming technique, which has to construct an accumulated matrix at first. The accumulated matrix  $D$  is constructed as following:

- 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 in Fig. 3.9 (c) is shown in Fig. 3.9 (d). The optimal

warping path is computed in reverse order of the indices starting with  $p_L = (N, M)$ . The pseudo code of the algorithm is shown below.

The optimal warping path computed by above algorithm is shown by red curve in Fig. 3.9 (c) (d), which indicates the optimal alignment between the two skeleton segments, as shown in Fig. 3.9 (a). Intuitively, such an optimal alignment runs along a “valley” of low cost within the cost matrix  $C$  and the accumulated matrix  $D$ . We can notice that there are two horizontal line segments on the optimal warping path in Fig. 3.9 (d), in which condition a single element of the first skeleton is aligned to many consecutive elements of the second skeleton, this result can be explicitly observed in Fig. 3.9 (a).

---

**Algorithm 3.1:** 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

```

---

This kind of alignment may cause the local feature of sequence take precedence over the global feature. Such an example is shown in Fig. 3.10 (a-c), in which the alignment between the skeletons of two scenario segments is computed by DTW. Intuitively, both of the two skeleton segments resemble an arch globally. However, the matched scenario segment searched from motion library which is similar to the query scenario segment has a relatively flat part in the middle, which causes a trough in the middle of the curvature curve as shown in Fig. 3.10 (c). The alignment result shows that the most part of the query scenario segment is aligned to the part of the matched scenario

segment with first wave crest on curvature curve, which appears as a long horizontal line in the optimal warping path on the accumulated matrix as shown in Fig. 3.10 (b). To avoid such degeneration, one common DTW variant is to impose global constraint conditions on the admissible warping paths. Such constraints do not only speed up DTW computations but also prevent pathological alignments by globally controlling the route of a warping path. More precisely, let  $R \subseteq [1:N] \times [1:M]$  be a subset referred to as global constraint region. Then a warping path relative to  $R$  is a warping path that entirely runs within the region  $R$ . One of well-known global constraint is the *Sakeo-Chiba band*, which is shown in Fig. 3.10 (e). Alignments of cells can be selected only from the respective shaded region. The *Sakeo-Chiba band* runs along the main diagonal and has a fixed width  $T \in \mathbb{N}$ . The *Sakeo-Chiba band* defined in Fig. 3.10 (e) restricts that an element  $x_n$  can be aligned only to one of the elements  $y_m$  with:

$$m \in \left[ \frac{M}{N} * n - T, \frac{M}{N} * n + T \right] \cap [1:M] \quad (3.7)$$

the  $T = 5$  in our application. The optimal warping path  $p_R^*$  in the constraint region  $R$  can be computed similar to the unconstrained case by formally setting  $c(x_n, y_m) := \infty$  for all  $(n, m) \in [1:N] \times [1,M] \setminus R$ . Under the global constraint, the new matched scenario segment searched from the motion library is shown in Fig. 3.10 (d), and the alignment of its curvature curve to the query scenario segment is shown in Fig. 3.10 (f).

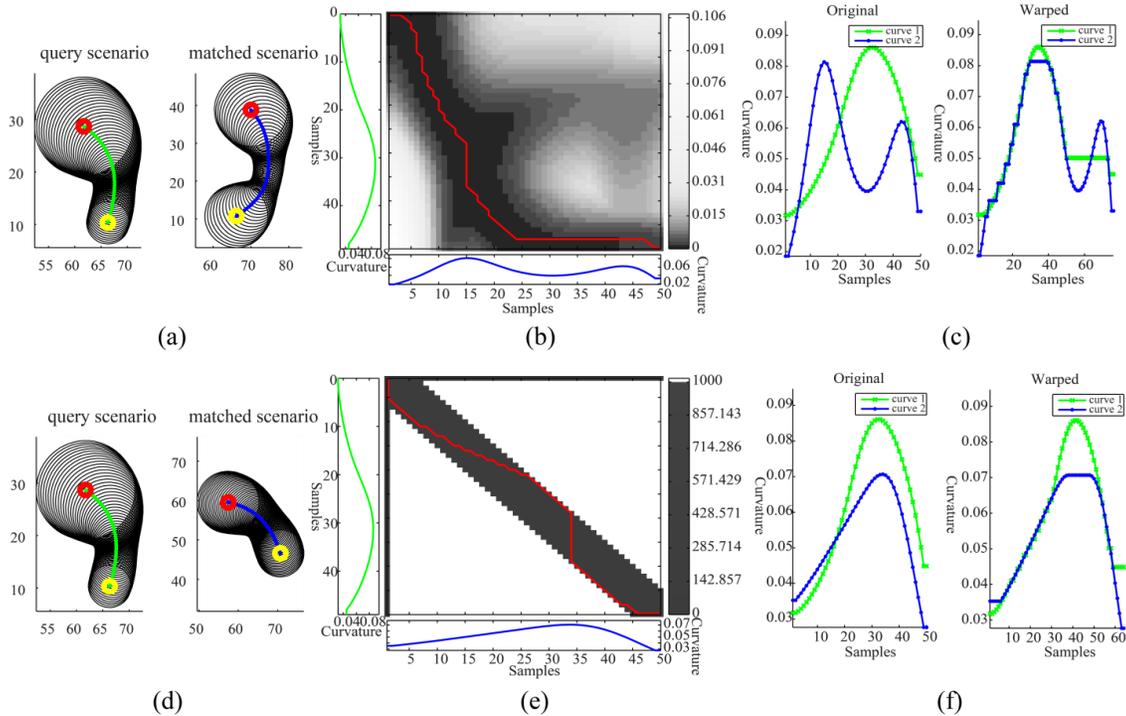


Figure 3.10. Imposing global constraint to limit the warping path

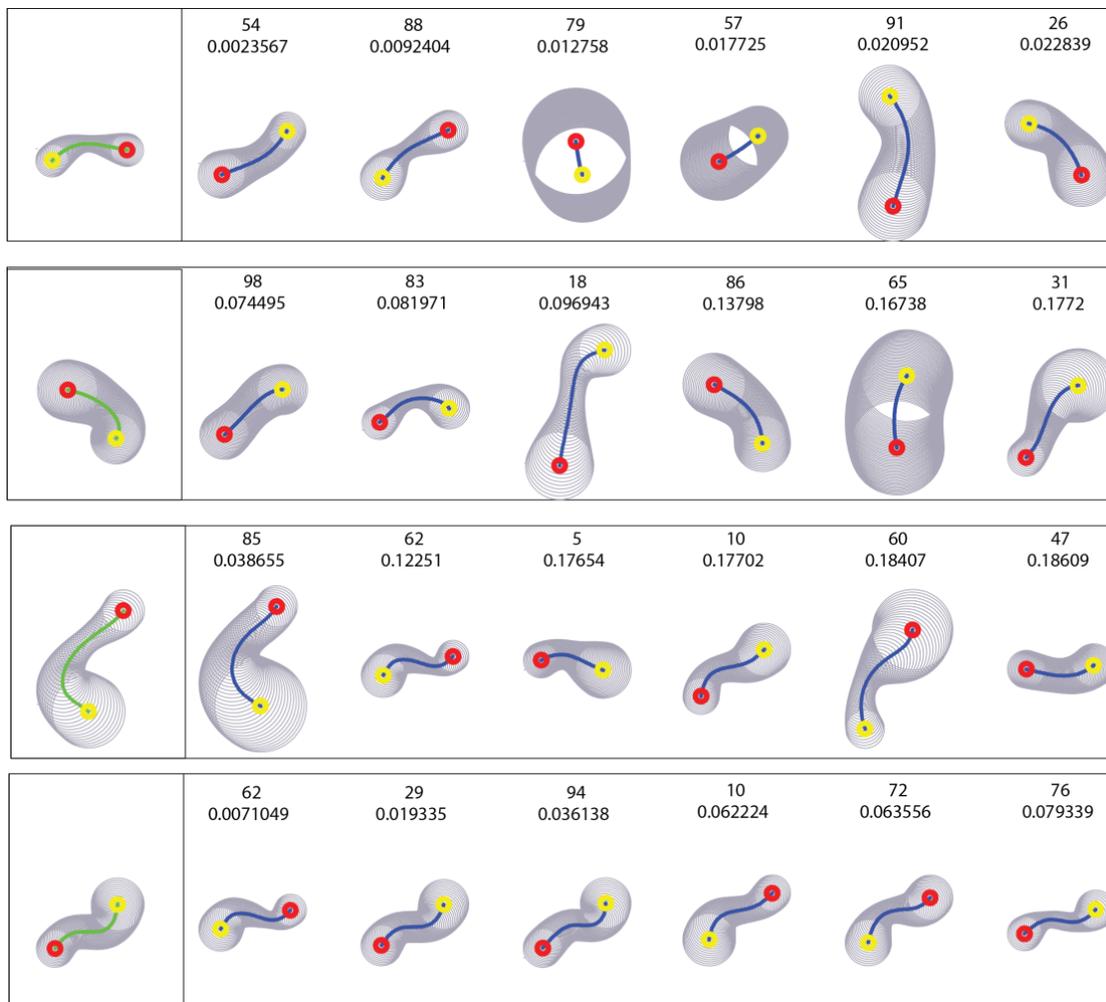


Figure 3.11. Sorted matching results for 4 query scenario skeleton segments by DTW using curvature feature

The further experimental results for skeleton matching by DTW using curvature feature are shown in Fig. 3.11. For each of 4 query scenario segments, we searched out the 6 most similar skeleton segments in a motion library with 100 segments, and sorted the results according to their similarity distance to the query segment, which is labeled on top of them with their number in the motion library. From the results, we can see that the event two skeletons have obvious topological similarity usually happen when their distance is less than 0.1. Moreover, two scenario segments with similar skeletons may have quite different radius changes, which may cause completely different movements some situations. Therefore, in order to find a completely similar scenario segment to a query scenario segment, we need to consider both curvature feature and radius feature.

When both of the two comparing scenario segments are narrow, the form of skeleton usually determines the movement pattern of the object.

### 3.5.3. Conclusion

According to the hypothesis of previous section, the similarity of scenario segments is measured by comparing the similarity of the sphere sequence's centerline and radius change. Since the centerline and radius change are two curves in their own space, the dynamic time warping (DTW) has been introduced to measure the similarity and compute the alignment of the two kinds of sequences. Although we only provide the examples in 2D, the method measuring the similarity of scenario segment can be easily extended to 3D. In 3D, the skeleton of a scenario segment is a curve in three-dimensional workspace. The curvature of a regular space curve is the magnitude of the acceleration of a particle moving with unit speed along a curve. The direction of the curvature in 3D is represented by a three dimensional normal vector which points to the center of the osculating circle from the contact point rather than the positive and negative values in 2D. Therefore, the curvature in 3D is represented as a triple  $(\alpha, \beta, \gamma) * \rho$ , in which the normal vector  $(\alpha, \beta, \gamma)$  and the parameter  $\rho$  indicate the direction and magnitude of the curvature respectively. The local cost  $c(x_n, y_m)$  between two elements of curvature curve can also be calculated by the Euclidean distance, and the rest computation of DTW runs as the same way as in 2D.

In this section, the Dynamic Time Warping is applied to compute the alignment of two scenario segment. However, this doesn't exclude the possibilities to apply some other algorithms in curve matching, time series retrieval and trajectory database filed. Our current work aims to verify the practicability of applying the curve pattern recognition method into the motion learning framework. Moreover, it is important to note that, the DTW metric does not obey the triangular inequality and thus has resisted attempts of exact indexing. In next section, we will provide the indexing structure of scenario segments and the method organizing the scenario segments with a hierarchical structure in motion library.

## 3.6. Organization of scenario segments in motion library

No matter how efficient the implemented motion planner is, we can finally get the conclusion about whether or not an objective part could pass through a concerned scenario in the workspace. In condition of feasibility, a collision-free motion trajectory of the objective part is obtained in the concerned scenario. Then, for this motion planning result, we extract the skeleton of the scenario, and divided the scenario into segments, each of which contains a segment of the scenario's skeleton and its corresponding motion trajectory. At the final phase of the learning process, we gather these scenario segments and their motion segments into motion library. It is also important to notice that

we also gather scenario segments which do not contain collision-free motions, because the failed scenario segments also provide the experience about which kind of scenario segments can be intuitively verified of non-accessible.

The motion library will gradually form into a large motion database with the increase of motion planning times. In motion generation process, we need make K-nearest neighbor search (K-NN) on motion library to find K most similar scenarios to a new scenario. If the simple sequential indexing method is adopted, this process would be very time-consuming. For efficiency reason, when gathering the scenario skeletons and their motion sequence, it is necessary to organize these scenario skeletons into a hierarchical tree structure. The indexing of sequential object has been widely studied since (Agrawal, Faloutsos et al. 1993), in which authors proposed using Discrete Fourier Transformation (DFT) to map a temporal sequence to a point in multidimensional space, every coordinate of which indicates the coefficient of a special frequency in the temporal sequence's frequency spectrum. Then these points are clustered with R-tree algorithm. Rather than Euclidean distance metric on the coefficients of DFT, DTW does not obey the triangular inequality and thus has resisted attempts at exact indexing. In (Keogh and Ratanamahatana 2005), authors studied the exact multidimensional indexing of dynamic time warping and proved the guarantee of no false dismissals of their method.

### 3.6.1. Indexing structure of scenario segments

The global constraint limits the warping path in a band along the diagonal path of the accumulated matrix. In the method of (Keogh and Ratanamahatana 2005), the authors define the *Upper* and *Lower* constraints to envelope the query curve, which is similar to a bounding filter. The idea behind this operation tends to use a bounding filter excluding unconcerned curves before using expensive computation of DTW. The *Upper* and *Lower* constraints are created according to the global constraint. We have introduced the global constraint *Sakeo-Chiba band*, which has constant width  $T$ , as shown in Fig. 3.12 (a). For an element  $x_n$  of the first sequence, the sequential number of  $y_m$  of the second sequence which can be aligned to the  $x_n$  is computed by Eq. 3.7. Since both of the  $N$  and  $M$  equals to fifty in our application, for the indices of a warping path  $(n, m)_l$ , we can get that  $m - T < n < m + T$ . Given the curvature curve  $Q := (q_1, \dots, q_l, \dots, q_L)$  of a scenario segment, the *Upper*  $U$  and *Lower*  $L$  constraints are defined as follows:

$$\begin{aligned} U_n &= \max(q_{n-T} : q_{n+T}) \\ L_n &= \min(q_{n-T} : q_{n+T}), n \in [1 : N] \end{aligned} \quad (3.8)$$

For the scenario segment in Fig. 3.12 (b), its *Upper* and *Lower* constraints are shown in Fig. 3.12 (c). Intuitively, the *Upper* and *Lower* constraints form a bounding envelope that encloses  $Q$  from above and below. Note that, although the *Sakeo-Chiba band* is of constant width, the corresponding

envelope generally is not of uniform thickness. In particular, the envelope is wider when the underlying query sequence is changing rapidly, and narrower when the query sequence plateaus.

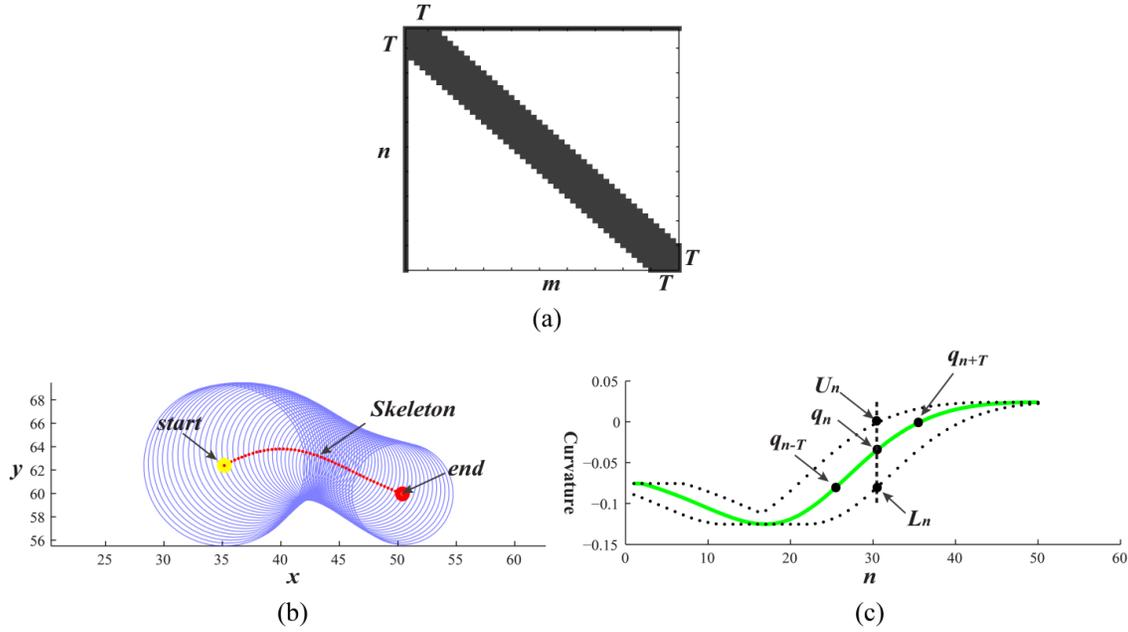


Figure 3.12. The *Upper* and *Lower* constraints on the curvature curve of a scenario skeleton

Based on the *Upper* and *Lower* constraints, Keogh and Ratanamahatana continued to propose using piecewise aggregate approximation (PAA) to transform a sequential curve into multidimensional representation, which make it possible to use multidimensional indexing structure like R tree for dynamic time warping. The PAA technique models a sequence by a linear combination of square wave pulses. For the given *Upper* and *Lower* constraints  $U$  and  $L$ , their piecewise aggregate approximations  $U_r$  and  $L_r$  are defined as follows:

$$\begin{aligned} (U_r)_n &= \max(U_{\frac{N}{K}^{(n-1)+1}} : U_{\frac{N}{K}^n}) \\ (L_r)_n &= \min(L_{\frac{N}{K}^{(n-1)+1}} : L_{\frac{N}{K}^n}), n \in [1 : K], 1 \leq K \leq N \end{aligned} \quad (3.9)$$

The piecewise aggregate approximation will reduce a sequence with length  $N$  into  $K$  dimensions. The dimension  $K$  is selected no more than 16, because multidimensional index structures begin to degrade rapidly somewhere above 16 dimensions. For the scenario segment in Fig. 3.13 (a), its piecewise aggregate approximation of *Upper* and *Lower* constraints are shown in Fig. 3.13 (b).

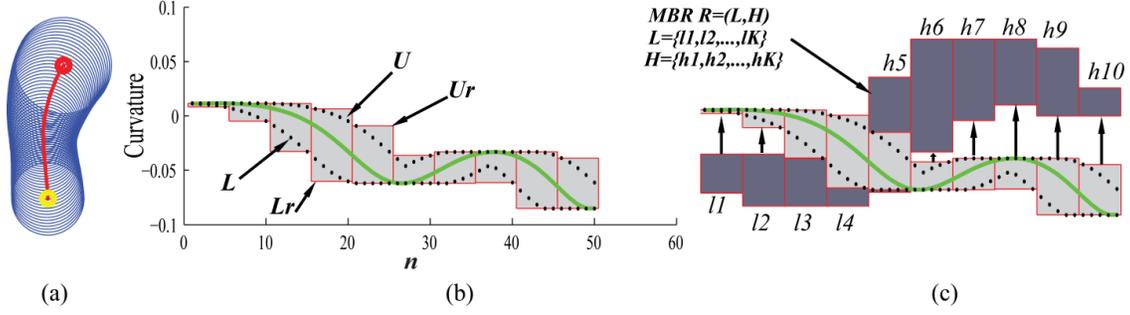


Figure 3.13. Piecewise aggregate approximation (PAA) of bounding constraints and the distance calculation between a query scenario segment and a minimum bounding rectangle.

For a multidimensional indexing technique, the indexing object is usually enclosed by a minimum bounding rectangle (MBR) which is contained in the leaf node of the hierarchical tree. When indexing, the decision about whether or not indexing the subtree of a node is determined by measuring the distance  $Dist$  between the query object and the MBR of this node. The piecewise aggregate approximations  $U_r$  and  $L_r$  can be regarded as enclosing a sequence by a MBR in multidimensional space. Therefore, suppose we have a node in the hierarchical tree, the MBR associated to this node is  $R$ . Let  $R = (L, H)$ , where  $L = (l_1, l_2, \dots, l_K)$  and  $H = (h_1, h_2, \dots, h_K)$ . Given the curvature curve of a query scenario segment  $Q$  and its piecewise aggregate approximations  $U_r$  and  $L_r$ , the distance  $Dist$  between  $Q$  and  $R$  is defined as follows:

$$Dist(Q, R) = \sqrt{\sum_{i=1}^K (d_i)^2}, d_i = \begin{cases} l_i - U_{r_i}, & \text{if } l_i > U_{r_i} \\ h_i - L_{r_i}, & \text{if } h_i < L_{r_i} \\ 0, & \text{otherwise} \end{cases} \quad (3.10)$$

The computation of the  $Dist$  between the piecewise aggregate approximations  $U_r$  and  $L_r$  of the query scenario segment in Fig. 3.13 (a-b) is shown in Fig. 3.13 (c), which is the sum and square root of the lengths of arrow lines.

### 3.6.2. R-tree introduction

Until now, a question has not been discussed about how many scenario segments are needed in motion library so that the experience is enough for a motion planning task in a new environment. In fact, this question is not easy to answer which is similar to asking how many motion skills that a child has to learn from his birth so that he can handle all the situations in real life. However, the fact we make sure is the learning runs through all one's life. Therefore, an ideal motion learning process should be able to dynamically update its experience according to the change of objective environment. R-tree and its variants are appropriate tools to organize the scenario segments in a dynamic and balanced hierarchical structure.

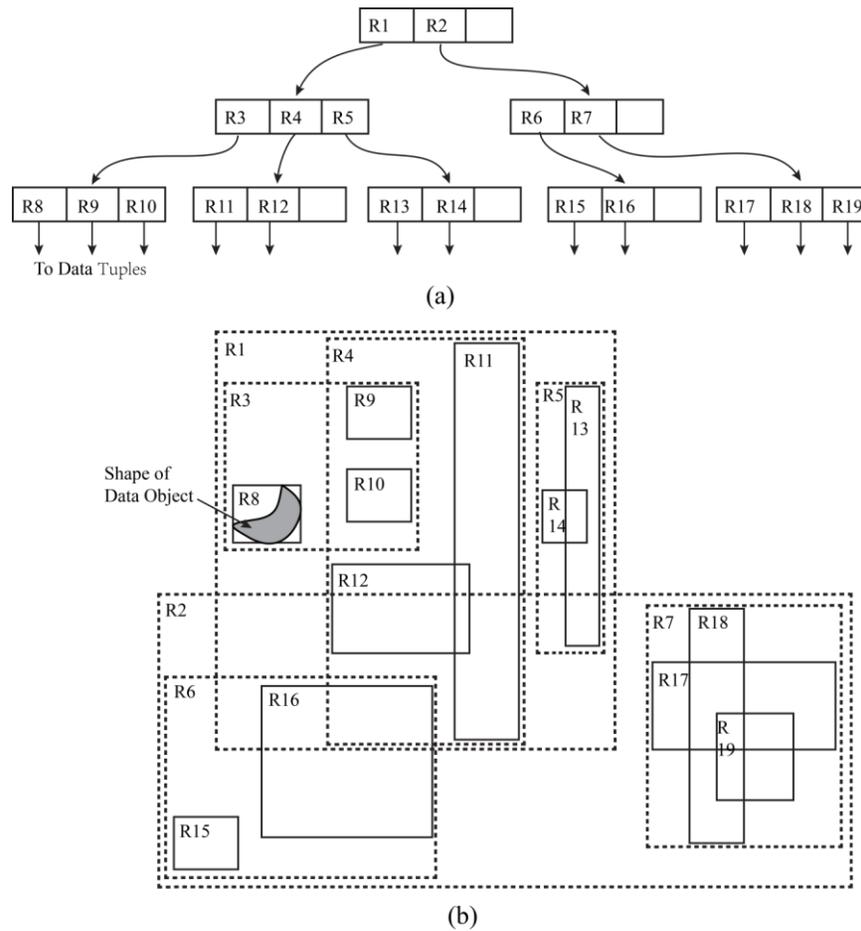


Figure 3.14. An example of R-tree (Guttman 1984)

R-trees are tree data structure used for indexing multidimensional information, which was proposed by (Guttman 1984). The main idea of R tree is organizing objects in multidimensional space into minimum bounding rectangles (MBR), higher level node in R tree is actually a MBR with larger size which containing many lower nodes (MBR) with smaller size. As shown in Fig. 3.14 (b), the minimum bounding rectangle  $R_8$  encloses the shape of an object. R-tree is also a balanced search tree, and supports dynamically inserting and deleting objects in the tree. The searching algorithms are rather simple which use the MBR to decide whether or not to search into a subtree.

In R-tree, the spatial objects are usually represented by tuples, and each tuple has a unique identifier which can be used to retrieve it. Leaf node in an R-tree contains index record entries of the form  $(I, \text{tuple} - \text{identifier})$ , where  $\text{tuple} - \text{identifier}$  refers to a tuple in the database and  $I$  is an  $n$ -dimensional rectangle which is the bounding box of the spatial object indexed  $I = (I_0 I_1, \dots, I_{n-1})$ . The  $n$  is the number of dimensions and  $I_i$  is a closed bounded interval  $[a, b]$  describing the extent of

the object along dimension  $i$ . Non-leaf node in R-tree contains the entries of the form  $(I, child - pointer)$ , where  $child - pointer$  is the address of a lower node in the R-tree and  $I$  covers all the rectangles in the lower node's entries. Let  $M$  be the maximum number of entries that will fit in one node and  $m \leq M/2$  specifying the minimum number of entries in a node. Fig. 3.14 shows the structure of an R-tree and illustrates the containment and overlapping relationships that can exist between its rectangles.

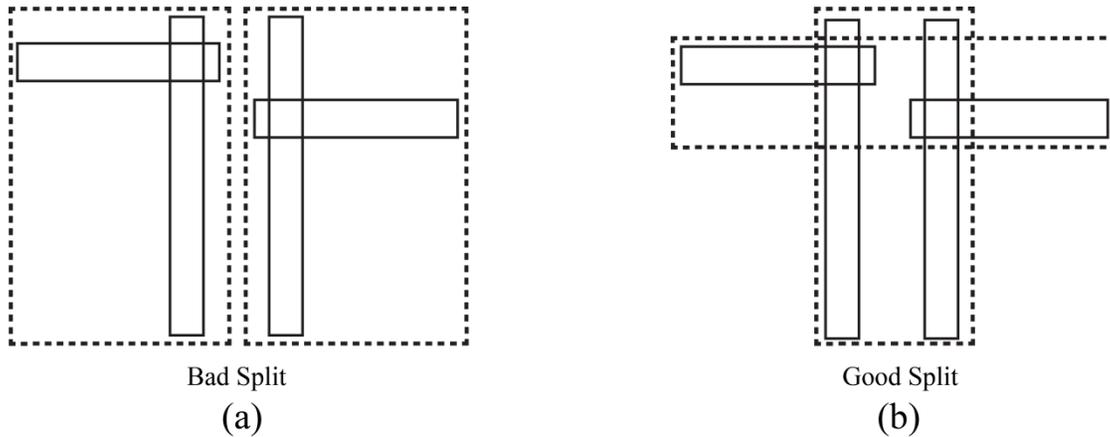


Figure 3.15. Bad and good split of an R-tree (Guttman 1984)

It is important to notice that all leaf nodes of R-tree appear on the same level. This property of R-tree is guaranteed by the insertion and deletion accompanied with dynamic management. If inserting a new data tuple into a leaf node, this node will split when it is overflow after insertion, the splits will propagate up to the root node. Guttman also discussed in his paper that a good split makes sure the total area of the two covering rectangles after a split should be minimized. Examples of bad and good splits are given in Fig. 3.15.

In all R-tree variants that have appeared in the literature, tree traversals for any kind of operations are executed in exactly the same way as in the original R-tree. Basically, the variations of R-trees differ in how they perform splits during insertion by considering different minimization criteria instead of the sum of the areas of the two resulting nodes (Manolopoulos, Nanopoulos et al. 2006).

### 3.6.3. Clustering scenario segments with R-tree

In our system, after we get a scenario segment and its motion segment, we generate  $(U_r, L_r)$  for the curvature curve of the skeleton segment. Using the method stated above, we find a leaf node on the tree of motion library, which is the best size MBR enclosing this query skeleton segment. Then save

the query skeleton segment into this MBR. If the MBR is overflow, it is split and the tree is updated according to R-tree rules. The clustering process is shown in Fig. 3.16.

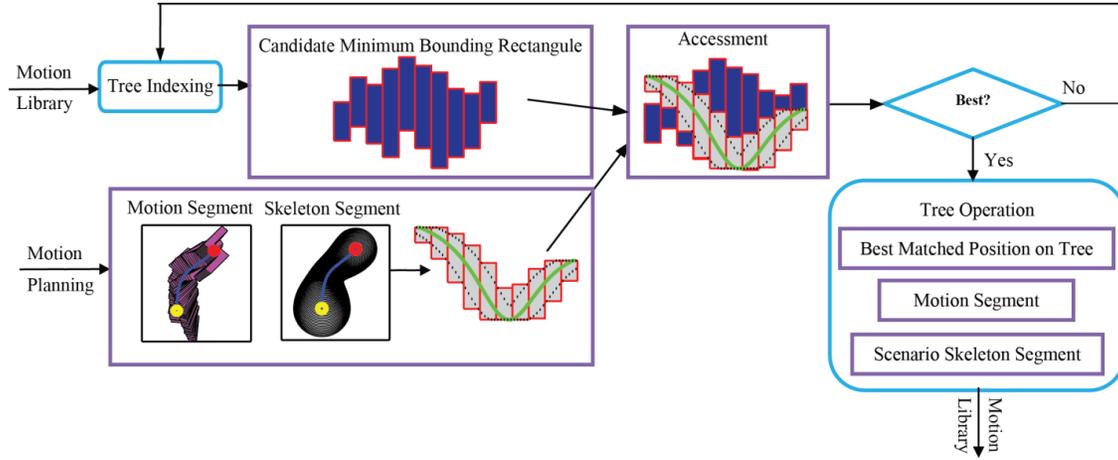


Figure 3.16. Process clustering scenario skeleton and its motion into motion library

Corresponding to basic R-tree, the piecewise aggregate approximations  $U_r$  and  $L_r$  encloses the entry scenario segment by a MBR, in which  $I_i$  of the rectangle is defined by  $[(U_r)_i, (L_r)_i]$ . Instead of computing the overlap between MBRs in basic R-tree, the distance between the query scenario segment and the indexing MBR is assessed by the Eq. 3.10. The deeper search is implemented in the subtree of MBR which has the shortest distance to the query scenario segment. Except the construction and distance measurement of MBR, the rest operations in motion library run in the same way as basic R-tree. In particular, the split of MBR in motion library also follows the same rule to make sure that the total area of the two covering rectangles after a split should be minimized. The split usually leads to the results which may differ from the perceptible experience. An example comparing the good and bad split of an R-tree in motion library is shown in Fig. 3.17. In this example, the maximum number of children  $M$  that a node can contain is set to 3. After an insertion, a node contains four children, which exceeds the number  $M$ . The MBRs of these four entries are  $R_3$ ,  $R_4$ ,  $R_5$  and  $R_6$ . There are three possible split strategies for basic R-tree (Manolopoulos, Nanopoulos et al. 2006):

- 1) **Linear Split.** Choose two objects as seeds for the two nodes, where these objects are as far apart as possible. Then consider each remaining object in a random order and assign it to the node requiring the smallest enlargement of its respective MBR.
- 2) **Quadratic Split.** Choose two objects as seeds for the two nodes, where these objects if put together create as much dead space as possible (dead space is the space that remains from

the MBR if the areas of the two objects are ignored). Then, until there are no remaining objects, insert the object for which the difference of dead space is assigned to each of the two nodes is maximized in the node that requires less enlargement of its respective MBR.

- 3) **Exponential Split.** All possible groupings are exhaustively tested and the best is chosen with respect to the minimization of the MBR enlargement.

For exponential Split, the number of all possible groupings is approximately  $2^{M-1}$ , which is not reasonable when  $M$  is large. In our application, Linear Split is adopted for experiment. Since there are a great number of Split strategies in literature, our discussion here does not focus on which one is the best for scenario segment retrieval but on the possibility that the R-tree and its variants can be applied for the efficient organization of scenario segments in the motion library. For a scenario segment, the MBR defines the range of bending degree of the segment's skeleton. The split strategy makes sure that the MBRs with similar range of bending degree are grouped together. As shown in Fig. 3.17, the bending degree of  $R_4$  is obviously much larger than  $R_3$ ,  $R_5$  and  $R_6$ . Grouping  $R_4$  with any one of  $R_3$ ,  $R_5$  and  $R_6$  will create a bad split. The good split groups  $R_3$ ,  $R_5$  and  $R_6$  together into new node  $R_2$  and assigns  $R_4$  into new node  $R_1$ . It is important to notice, the scenario segments contained in a node may appear with quite different bending patterns but with similar range of bending. The associated explanation is the speed of translation and rotation of object is positively correlated with the bending degree of scenario skeleton. Therefore, the scenario segments with quite different bending degree to the query scenario segment are not expected to be visited in the retrieval process.

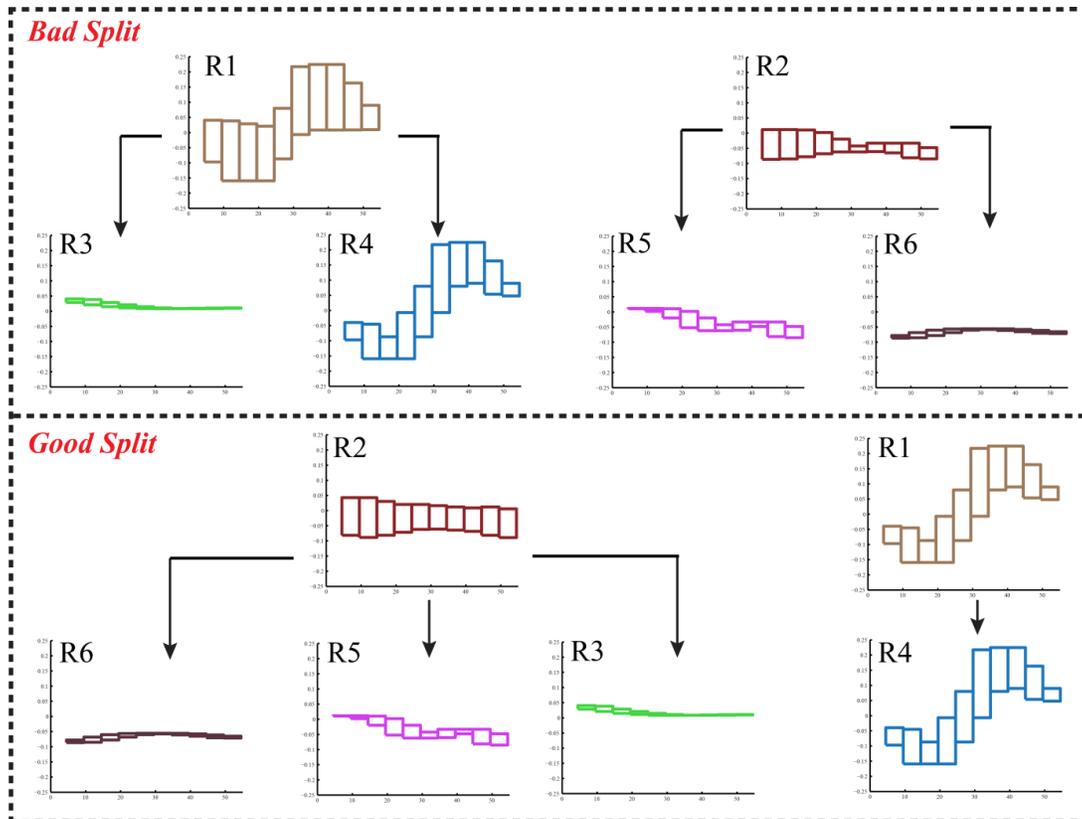


Figure 3.17. Good and bad split of an R-tree in motion library

The clustering process will organize the scenario segments into a multidimensional tree structure. As this tree structure is a multi-tree, most of the scenario segments will not be visited in a single searching process. Moreover, the searching until leaf nodes only calculates the distance between  $(U_r, L_r)$  and MBR by Euclidean metric, it will be very efficient for K-NN search in scenario retrieval process. In the motion library, we cluster the Successful Scenarios (scenario segment with collision-free motion sequence) and Failure Scenarios (scenario segment without collision-free motion sequence) into two independent trees respectively, named as Successful Tree and Failure Tree. The Failure Scenarios indicate the situations in which the accessibility for the concerned object is untenable. The retrieval on Failure Tree is useful for detecting the inaccessibility of a new scenario before a complete motion planning. The scenario segments recorded in the motion library are also marked by its visiting frequency. The seldom visited scenario segment will be deleted from the motion library. Fig. 3.18 demonstrates an R-tree constructed from ten scenario segments.

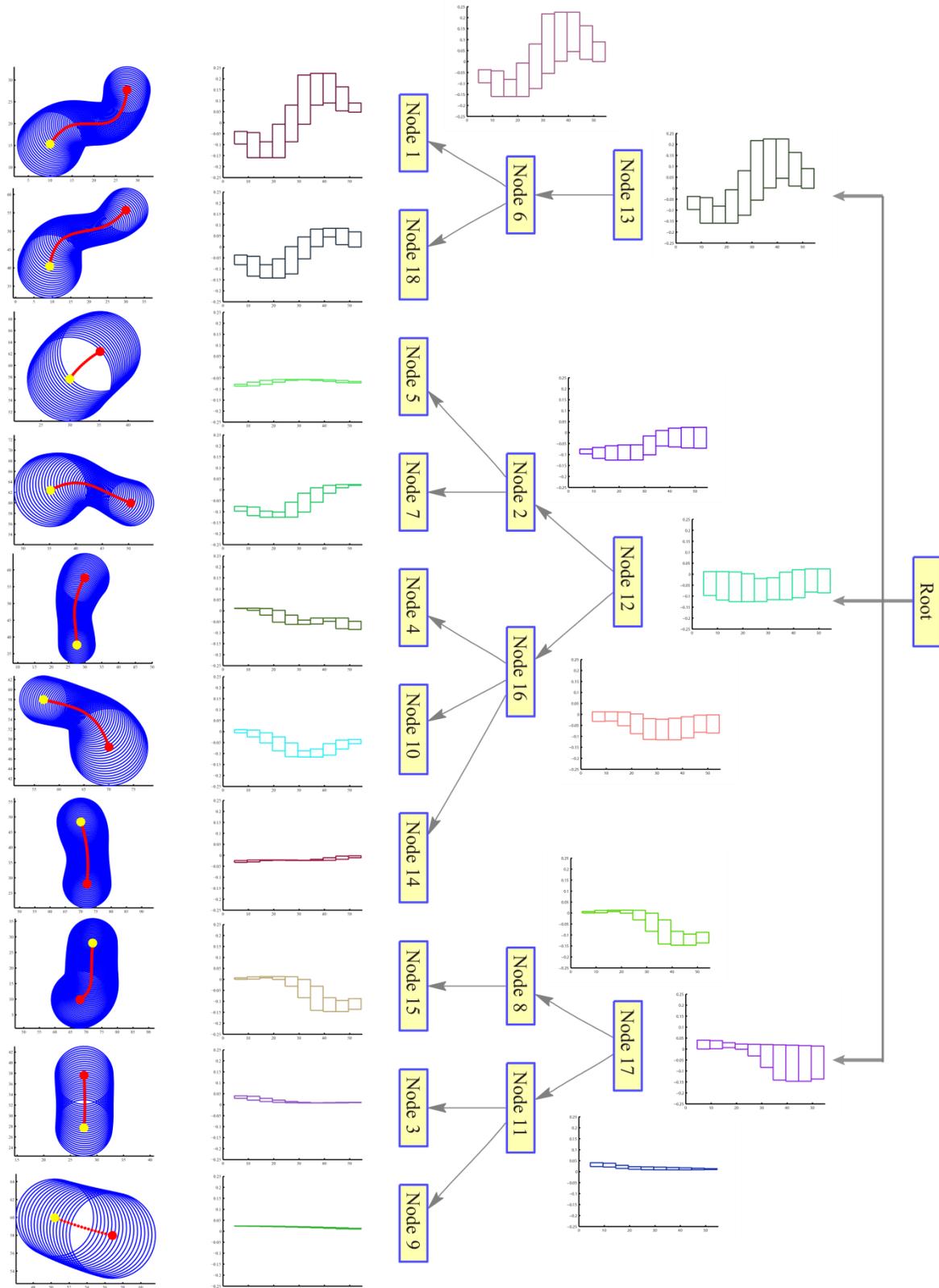


Figure 3.18. An R-tree constructed from ten scenario segments

Fig. 3.19 shows the searching time in the motion library with the R-tree structure, in which  $M$  stands for the maximal number of children in a non-leaf node and  $N$  stands for the total number of scenario segments in the motion library. We can see that the search in the motion library with 100 scenario segments is less than half millisecond in Matlab.

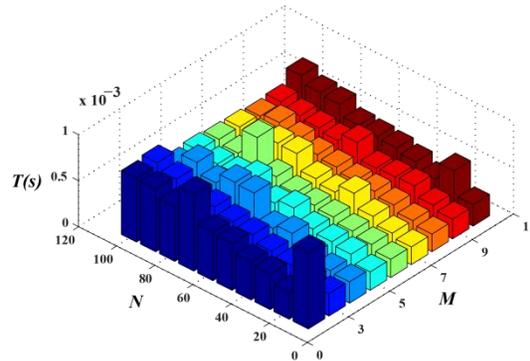


Figure 3.19. Searching time in motion library with the R-tree structure

### 3.7. Conclusion

In this chapter, we firstly propose a hypothesis which assumes the movement pattern (the speed of translation and rotation) of object has positive correlation with the topological structure and volume change of the object's scenario. Then the scenario is represented by a sphere sequence, of which the centerline and radius change approximately describe the topological structure and volume change of the scenario. After segmentation, the scenario and its containing motion trajectory are split into segments. Based on the similarity measurement of scenario segments' skeleton, the scenario segments and their containing motion segments are recorded into motion library. Thanks to a specially designed indexing structure, the scenario segments are organized by R-tree in motion library. The discussion about how to use these scenario segments and motion segments for motion planning in an accessibility verification task will be discussed in the next chapter.

Although there remains a lot of work to extend this motion learning method to 3D application, the extension doesn't involve difficult technique issues. At first, the sphere sequence generation method is not difficult to be realized in 3D situations. The collision detection methods for CAD model such as PQP (Gottschalk, Lin et al. 1996) can compute the distance between the closest pair of points on two models. Therefore, the minimum distance between the reference point and the obstacles can be computed as well as in 3D. The maximal sphere around this reference point can be found out by a binary search which fixes the tangent point and enlarges the radius of the sphere until the sphere tangent to another point of the obstacles. The computation of curvature and similarity measurement in 3D has been talked about in section 3.5.3. For the extension of indexing structure, given the

curvature in 3D is represented as a triple  $(\alpha, \beta, \gamma) * \rho$  as the same definition in section 3.5.3, the Upper and Lower constraints are defined in the similar way to  $U(U_\alpha, U_\beta, U_\gamma)$  and  $L(L_\alpha, L_\beta, L_\gamma)$ , and the piecewise aggregate approximations  $U_r$  and  $L_r$  can be extended to  $U_r(U_{r\alpha}, U_{r\beta}, U_{r\gamma})$  and  $L_r(L_{r\alpha}, L_{r\beta}, L_{r\gamma})$ . Thus the MBR of curve in 3D situation is similar to a sequence of bounding box, this method can refer to (Vlachos, Hadjieleftheriou et al. 2003, Vlachos, Hadjieleftheriou et al. 2006).

There are still some limitations about extending our motion learning method into 3D situations. Our hypothesis is established on the situations in which the motion paths of all the points on an object are enclosed by the scenario. Our method does not accommodate the situations in which the objects can be clamped to each other, such as the situation of famous alpha puzzle benchmark, because the sphere sequence is not enough to represent the relationship between the movement of object and its scenario.

# 4 Interactive Motion Planning with Learned Experience

---

4.1. Introduction .....	89
4.2. Motion generation process .....	90
4.2.1. Scenario selection.....	91
4.2.2. Motion retrieval.....	95
4.2.3. Motion transformation .....	97
4.2.4. Motion repair.....	98
4.3. Application results .....	98
4.3.1. Results in feasible scenarios.....	99
4.3.2. Results in non-feasible scenarios .....	104
4.4. Extension .....	106
4.4.1. Extension to the motion planning of articulated object.....	106
4.4.2. Extension to the differential motion planning .....	110
4.5. Conclusion.....	114

---

## 4.1. Introduction

In the last chapter, the scenario segments and motion segments are gathered from the previous motion planning process in the accessibility verification tasks. And these scenario segments and motion segments are organized into a hierarchical structure in the motion library based on their similarity. According to the hypothesis in the last chapter, the motion segment linked to a scenario

segment can be applied to some similar scenario segments. The similarity refers to the bending degree of scenario skeleton and the volume changing trend of the scenario segment. Therefore, for an unknown scenario in a new verification task, a strategy of reusing the motion information recorded in the motion library is to efficiently retrieve the similar scenarios and to reasonably transform the recorded motion into new valid motions for serving new task.

In this paper, a novel motion planning framework is proposed in (Yan, Poirson et al. 2015), which can use the recorded motion information introduced in the last chapter. A simple overview on the planning process can refer to the Fig 4.1.

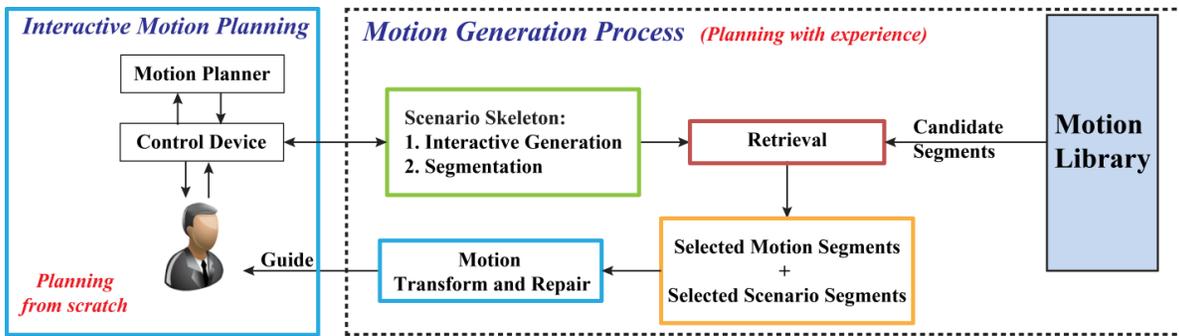


Figure 4.1. Overview of motion generation process

## 4.2. Motion generation process

The data flow of the motion generating process is described in fig. 4.2. At the initial phase of motion generation process, users can interactively select their interested scenario by manipulating a reference point moving in the workspace. According to the location of the reference point, the local maximal sphere is generated by the method in Section 3.4.3. The interesting scenario is the scenario which user wants the part to pass through. Since the interaction process only involves translational manipulation, it can greatly reduce the difficulties of previous interactive motion planning methods and 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 in the motion library are retrieved. The retrieval process will be firstly implemented on the Successful Tree. The retrieved motion segment is transformed to fit new scenario segment according to the alignment between scenario skeletons. If the transformed motions satisfy the condition with 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. After these operations, if a new collision-free motion segment is obtained in every new scenario segment, the following process connects the pairs of adjacent motion segments by automatic motion planners. The connection between motion segments runs at great speed, which is always negligible.

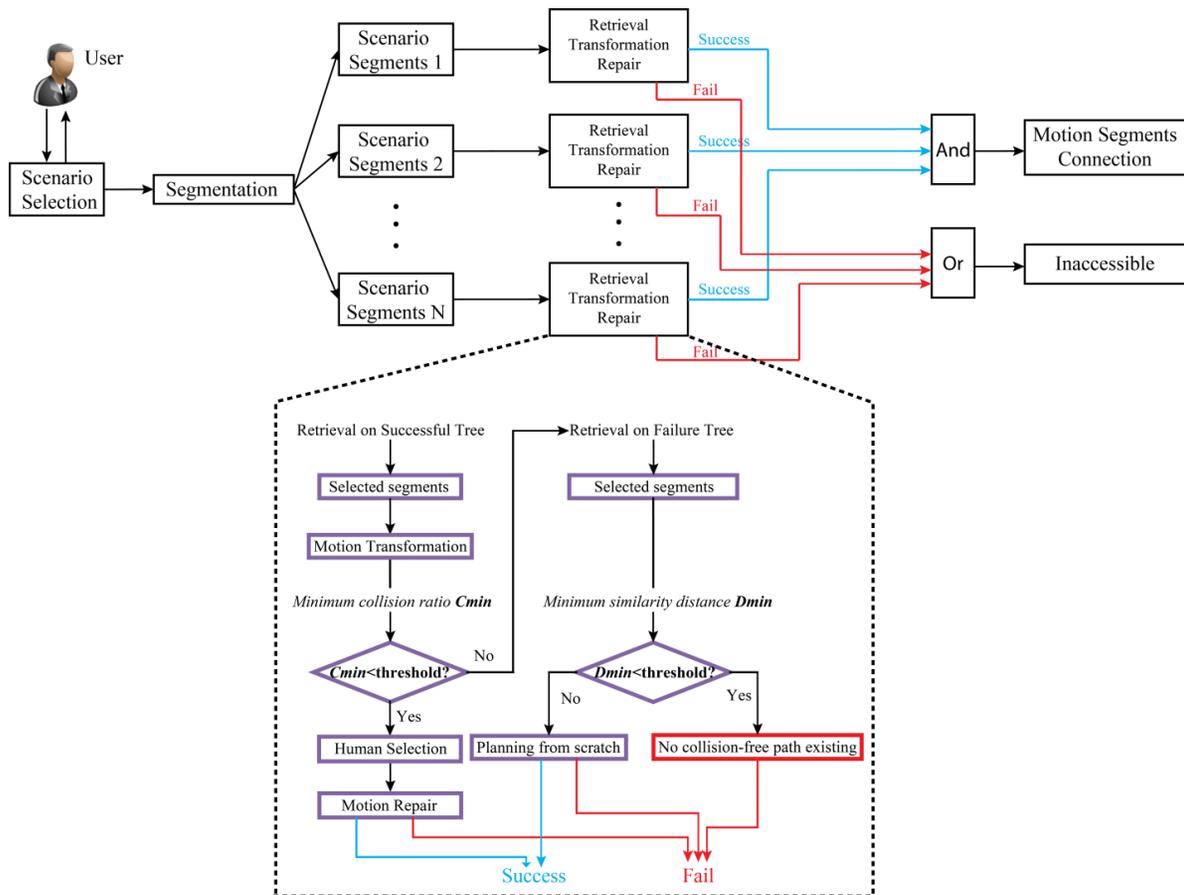


Figure 4.2. Motion Generation Process

#### 4.2.1. Scenario selection

For the interaction of a virtual motion planning task, an experience is that human has the advantage on the global planning ability. In some situations, human exactly knows which passages are necessary when the manipulated part is expected to have a continuous collision-free path between its initial location and the final location. However, human's global planning ability is limited by the

accurate manipulations in some complex situations. For example, in a cluttered assembly task, users usually know where to go but don't know how to go due to the local geometric constraints imposed by other parts. The previous interactive motion planning methods attempt to solve this problem through allowing automatic motion planners to provide local manipulation guidance to the user. However, when there are narrow passages in the part's collision-free configuration space  $\mathcal{CS}_{free}$ , the automatic algorithms are not enough to provide effective guidance. Thus, we try to provide a new paradigm of interaction. As shown in Fig. 4.2, the user is only asked to select the scenarios in the workspace. During the selection, the user needs to control only one moving point in the workspace. The controlled point is regarded as the reference point of method in Section 3.4.3. According to the trace of the controlled point, the maximal sphere sequence is created. There are three modes which can find the trace of the controlled point:

- 1) User generates: in this mode, the moving point is only controlled by user. In 2D workspace, a collision-free trajectory for a point object from the initial location to the final location is easy to be found by users. In 3D workspace, this is slightly more complex. Since some part of the workspace is not visible to the user, the trajectory is not always obvious and the controlled point can go into the dead end, like the solid curve in Fig. 4.3 (a).
- 2) Automatic generation and then selection by user: in this mode, the trajectory of the controlled point is firstly generated by the automatic path planners. The generated trajectory possibly translates through some passages which is infeasible for the object to pass through. Such as the situation in Fig. 4.3 (b), the trajectory (solid curve) of the controlled point generated by automatic path planner contains an invalid part (red band). Some of these invalid parts are obvious to users. Thus the user can restart another search of automatic path planner until he gets a satisfactory trajectory of the controlled point.
- 3) Generation by user with the guidance of automatic algorithms: to avoid the weakness of above two modes, a possible solution uses the automatic path planner to guide the user's manipulation. The objective is to allow user freely controlling the point to select his preferred scenario of workspace, and at the same time guarantee the controlled point does not go into the dead end or translate through some obviously infeasible passages.

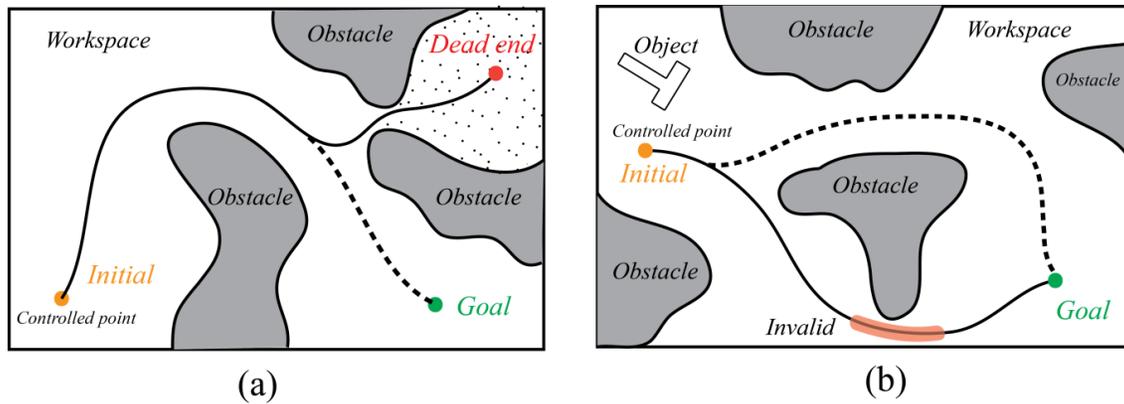


Figure 4.3. Two invalid situations during the generation of the controlled point's trajectory

The third mode is adopted in our method to find the trace of the controlled point. There are three basic rules for designing this interaction strategy. The first one is that, the automatic path planners should provide enough guidance to the user so that the manipulation of user will not fall into the dead end situations. The second rule is that the automatic path planner should not limit the freedom of user's selection. The third rule is that user's manipulation should have prior search in large volume of the workspace, because the regions with large volume have more chances to be feasible for the object than the regions with small volume.

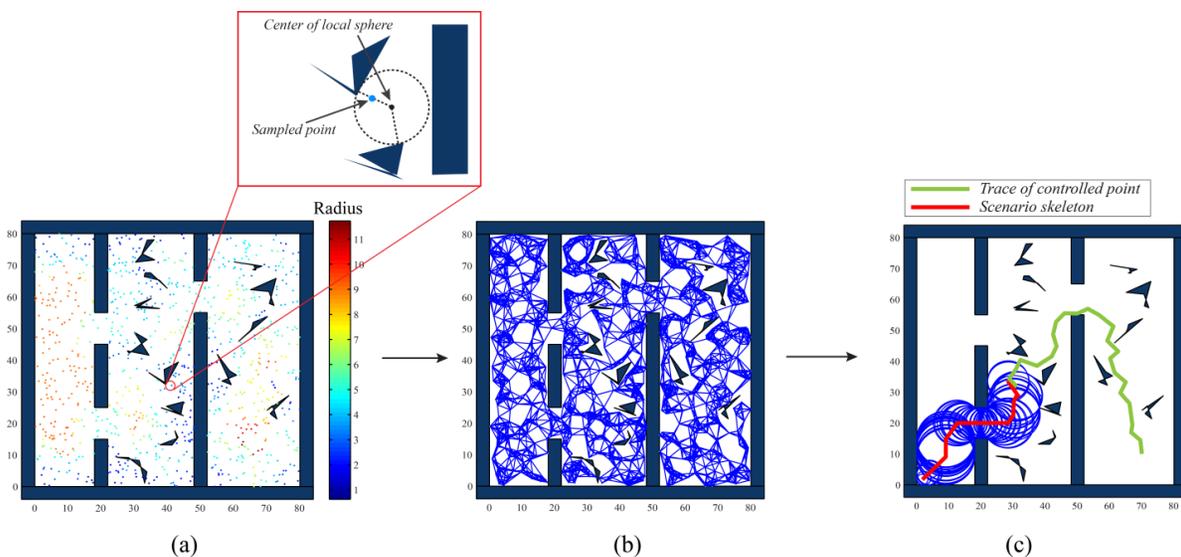


Figure 4.4. Modified PRM is used to generate the trace of the controlled point

Specifically, the modified PRM (Probabilistic Roadmap Planner) is used to guide user's manipulation during the generation of the trace of the controlled point. Recalling the basic PRM, it

consists of two phases: a construction phase and a query phase. In the construction phase, firstly the collision-free configurations are sampled in robot's configuration space  $\mathcal{CS}$ . In our application, the configuration space of the controlled point is the subset of Cartesian space. The basic PRM is modified here to mark the randomly sampled points in workspace with the radius of the surrounding local maximal sphere. Here, the randomly sampled point is taken as the reference point when using the method of Section 3.4.3 to create the local maximal sphere. As shown in Fig. 4.4 (a), the colors of the randomly sampled points in workplace are marked according to the radius of the surrounding local maximal sphere.

After random sampling, the created points which lie outside of the obstacles are connected to the adjacent points with the valid straight lines which do not collide with the obstacles. This process has been demonstrated in the Fig. 4.4 (b). After connection, a roadmap which captures the connectivity of the workspace for a point robot has been constructed.

After the construction phase, if a goal location is defined, a collision-free path for the controlled point from its current location to the goal location can be found out on the constructed roadmap with the Dijkstra algorithm in real time. In particular, our method calculates the length of the roadmap edge by the radius of this edge's two end points' surrounding spheres. Given two points  $P_1$  and  $P_2$  of the roadmap, the radius of their surrounding spheres is  $R_1$  and  $R_2$  respectively, the length of edge  $P_1P_2$  is defined as  $\alpha/(R_1 + R_2)$ , where  $\alpha$  is a constant, as shown in Fig. 4.5. This modification will result in that the final path of the controlled point found by PRM has bias to translate through the region of workspace with large volume.

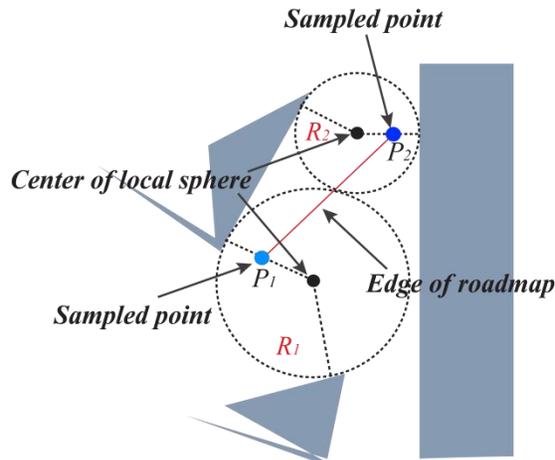


Figure 4.5. The computation of the length of roadmap edge

With the modified PRM algorithm, the user is allowed to control one point to search the interesting scenario of the workspace with the real time guidance, which tells him the right direction leading to the goal location with the guarantees of the prior search in the large volume. According to the trace of controlled point, the local maximal sphere sequence is created to represent the selected scenario. A case of this process is shown in the Fig. 4.4 (c).

#### 4.2.2. Motion retrieval

*Retrieval on the Successful Tree:* The selected scenario is split into segments by the method of Section 3.4.4 according to the change of the sphere radius. For each scenario segment, the retrieval process firstly makes a K-NN search in the motion library using the curvature feature of the skeleton. In the last chapter, the scenario segments have been organized by the R-tree structure in the motion library. The node on the tree is represented as a Minimum Bounding Rectangle (MBR) which encloses the node's children. The search outside MBR will calculate the distance between MBR and the  $(U_r, L_r)$  of the scenario skeleton's curvature curve. The search inside MBR will use DTW to calculate the distance. After K-NN search,  $k$  scenario segments with the most similar bending trend to new scenario segment are retrieved. Then, the distance between each of the  $k$  scenario segments and new scenario segment is measured by DTW using the radius feature. In the sorted result, the five highest ranking scenario segments are selected and output to the motion transformation module. This retrieval process is shown in the Fig. 4.6. Fig. 4.7 shows the ten retrieved scenario segments which have the similar bending skeleton to a query scenario segment. The five scenario segments selected from this retrieval result are shown in Fig. 4.8. In the two figures, the scenario segments are sorted by dissimilarity distance in ascending order.

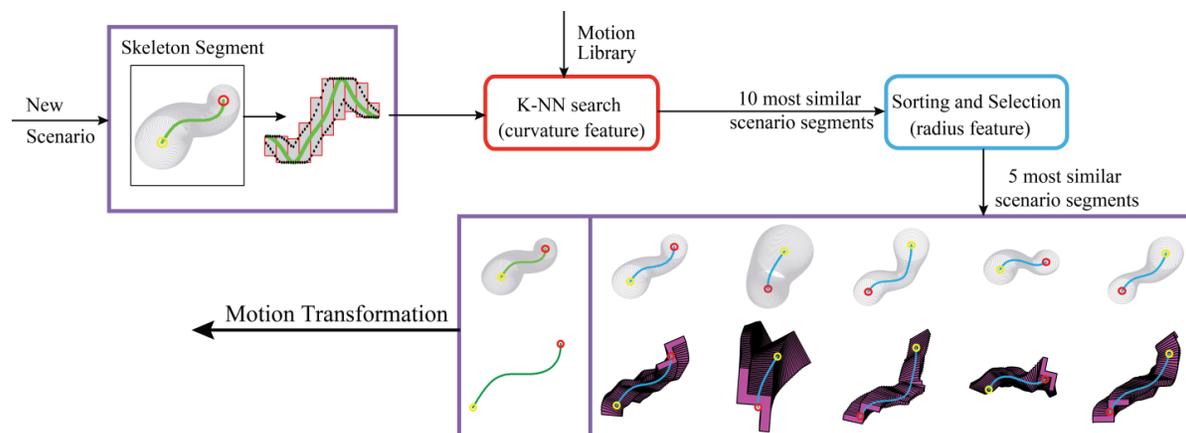


Figure 4.6. Scenario retrieval process

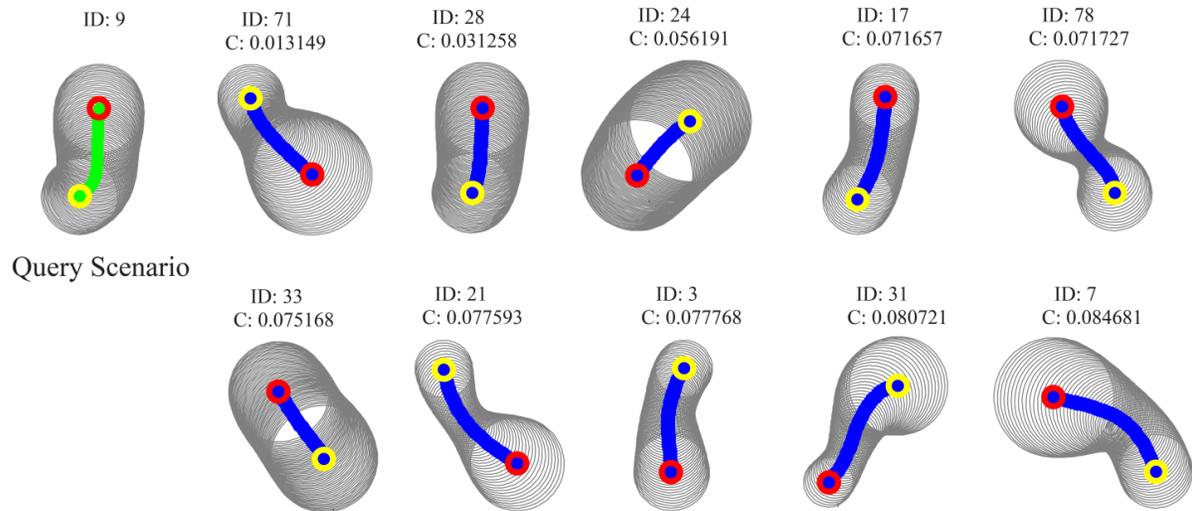


Figure 4.7. The ten retrieved scenario segments with similar skeleton to the query scenario segment (ID is the identifier number in the motion library; C is the dissimilarity distance of the curvature of scenario skeleton)

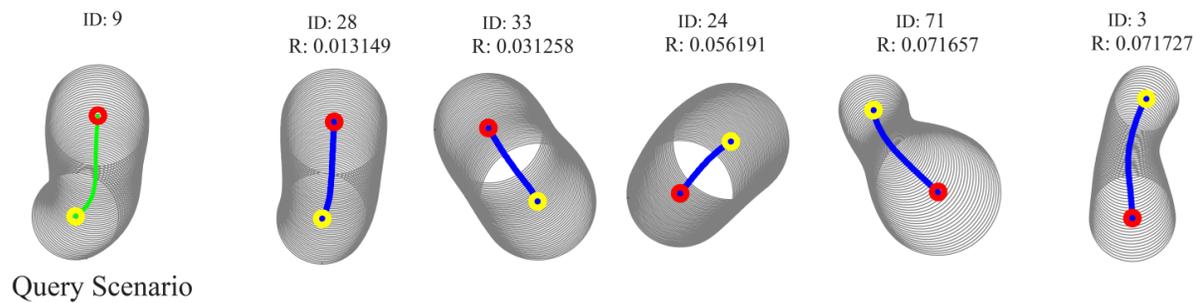


Figure 4.8. The five scenario segments selected from the ten retrieved segments (ID is the identifier number in the motion library; R is the dissimilarity distance of the radius of scenario)

*Retrieval on the Failure Tree:* If none of the retrieved motion segments can be used to produce a collision-free motion in the new scenario segment, the retrieval process will be started on the Failure Tree. The failure of the retrieved motion segments from the Successful Tree might result from two situations: one is that the motion library is not large enough, in which situation the new motion is planned from scratch; the other is that the new scenario segment is not feasible for the objective part to pass through. Thus the retrieval is continued on the Failure Tree to exclude the latter reason. The search method on the Failure Tree is the same as that on the Successful tree. If the similarity between new scenario segment and one of the returned segments is less than a threshold defined both by curvature and radius, it can be concluded that this new scenario is impossible to pass through.

## 4.2.3. Motion transformation

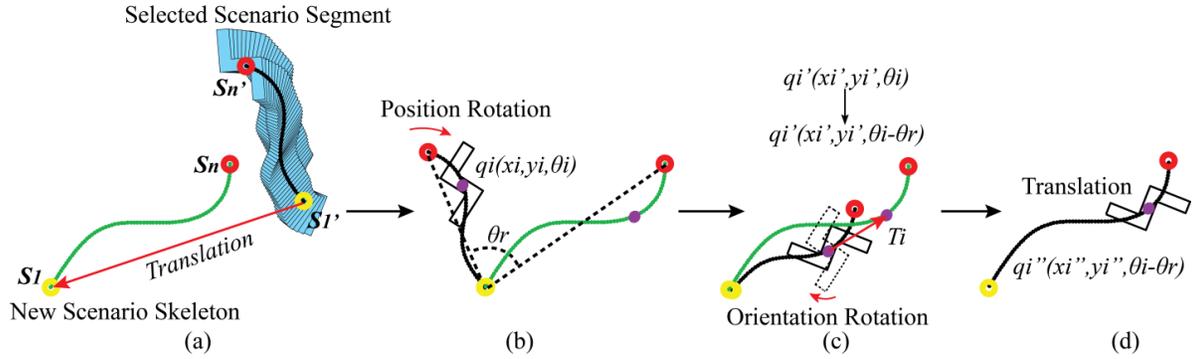


Figure 4.9. Motion transformation according to the skeleton transformation

If the candidate scenario segment is retrieved from the Successful Tree, it contains a motion segment, which is quite different in length, location and orientation in motion library. The motion segment is transformed in this phase according to the alignment between two scenarios' skeletons. Given the skeleton of 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 situation,  $s_i, s'_i \in \mathbb{R}^2$ ,  $q_i = (x_i, y_i, \theta_i) \in \mathbb{R}^2 \times SO(2)$ ; and in 3D situation,  $s_i, s'_i \in \mathbb{R}^3$ ,  $q_i = (x_i, y_i, z_i, \varphi_i, \psi_i, \omega_i) \in \mathbb{R}^3 \times SO(3)$ . The transformation of skeleton and motion are in space of different dimensions, thus the transformation matrix of skeleton cannot be directly used to motion segment. The transformation of a motion segment takes the following steps, see the example in 2D Fig. 4.9:

- 1) As shown in Fig. 4.9 (a), the initial step will firstly translate  $Sk_{sel}$  towards  $Sk_{New}$  and make their starting points (yellow) overlap. The translation vector is  $\overrightarrow{s'_1s_1}$ . Then the position coordinates of every configuration in  $M_{sel}$  do the same translation along vector  $\overrightarrow{s'_1s_1}$ . Update  $Sk_{sel}$  and  $M_{sel}$ .
- 2) After initial translation, an angle  $\theta_r$  between  $\overrightarrow{s_1s_n}$  and  $\overrightarrow{s'_1s'_n}$  is got, as shown in Fig. 4.9 (b). Then the skeleton  $Sk_{sel}$  rotates towards  $Sk_{New}$  around its starting point  $s'_1$  by  $\theta_r$  angle, which makes  $\overrightarrow{s'_1s'_n}$  have the same orientation with  $\overrightarrow{s_1s_n}$ . 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 dashed line in Fig. 4.9 (c).
- 3) After position rotation, the configuration  $q_i'$  in  $M_{sel}$  has the wrong orientation. Then the objective part on  $q_i'$  rotates around its center of mass by  $\theta_r$ , the  $q_i'$  become  $q_i'' = (x_i', y_i', \theta_i - \theta_r)$ .

- 4) After previous transformations, the skeletons  $Sk_{sel}$  and  $Sk_{New}$  still mismatch in Fig. 4.9 (c). Given  $s'_i$  of  $Sk_{sel}$  and its corresponding point  $s_j$  of  $Sk_{New}$ , the translation vector of  $s'_i$  is  $\vec{T}_i = \overrightarrow{s'_i s_j}$ . The pair of  $(i, j)$  is determined by the alignment between  $Sk_{sel}$  and  $Sk_{New}$ . The position coordinates of every configuration in  $M_{sel}$  do the same translation along its skeleton point's translation. For example,  $q_i'$  translate along  $\vec{T}_i$  to  $q_i'' = (x_i'', y_i'', \theta_i - \theta_r)$ , where  $x_i'' = x_i' + \vec{T}_i(x)$  and  $y_i'' = y_i' + \vec{T}_i(y)$ .

The transformation cannot guarantee the collision-free motion even though the skeletons of the new scenario and the selected scenario are very similar. In the implementation, a *Collision Ratio*  $q_{collision}$  is set, which is the ratio of  $N_{collision}$  to  $N_{total}$ . In new motion segment,  $N_{collision}$  is the number of in-collision configurations and  $N_{total}$  is the total number of configurations. If the *Collision Ratio*  $q_{collision}$  is larger than a threshold, this motion segment is considered as a 'non-satisfactory' motion; otherwise it is 'satisfactory'. If none of the selected segments is 'satisfactory', the retrieval process will be started on Failure Tree as stated in the previous section; otherwise the user is allowed to select a 'satisfactory' motion segment which is close to his preference.

#### 4.2.4. Motion repair

If the transformed motion segment contains some in-collision configurations, a motion repair phase will continue. Firstly, 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 *Generalized Penetration Depth* which is proposed in (Zhang, Kim et al. 2007). However, the optimality of generalized penetration depth is not necessary in our application, thus a lower cost algorithm in Section 2.3 is adopted for the retraction step. After the retraction, the adjacent configurations are connected with RRT-Connect algorithm. If all of the connections are successful, a collision-free motion sequence will be found in the new scenario segment.

### 4.3. Application results

In the following section, our motion planning framework will be applied to a non-convex part moving in a cluttered 2D environment which contains randomly shaped and located non-convex obstacles. The implementation results will demonstrate the efficiency of our method especially in narrow passages. Moreover, after adequate learning, the interaction in the planning process will release users from rotation manipulation, which greatly reduces virtual manipulation's dependence on the haptic device.

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

*User manipulation:* For the interactive motion planning module in Fig. 4.1, the method in the second chapter is adopted, which allows users to manipulate the object with some penetration into obstacles. In motion generation process, user can control a reference point in the workspace, for every position of which the maximal sphere tangent to the neighboring obstacles is generated in real time. The distance between two 2D polygons is computed by the method in (Jacquenot 2008). The controlled point can be manipulated by joystick or mouse. At the same time, the point isn't allowed to come up against obstacles. In this way, users can freely choose the region of workspace that he is interested in. For the Z shape part, the accessibility of some scenarios cannot be detected visually. In Fig. 4.10, a scenario translating through the environment is selected by users, the skeleton of which is marked as dashed curve. The results of two segments of the scenario (outlined by blue color) are analyzed in the following section. Before the experiments, the learning process ran in some similar environments and gathered 100 Successful Scenario segments and 75 Failure Scenario segments in motion library.

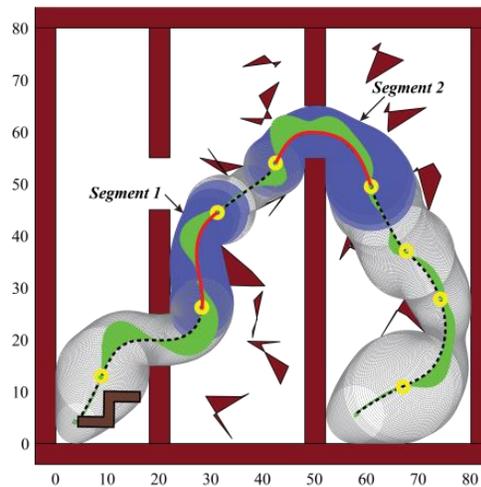


Figure 4.10. Two feasible scenario segments selected in an interaction

#### 4.3.1. Results in feasible scenarios

The results for the first selected scenario segment are demonstrated in Fig. 4.11. The sequential search is implemented to compute the similarity of scenario skeleton by DTW. The new scenario segment is compared to every scenario segment both in positive and reverse order. Here the Successful Scenario segments haven't been organized into R-Tree structure, because the sequential search in 100 scenario segments by DTW is fast enough for our applications. Ten best retrieval results are sorted through computing radius similarity by DTW, and the first 5 scenario segments are selected, which are shown in Fig. 4.11 (a). For these 5 scenario segments, their ID number in the motion library and their radius and curvature similarity to new scenario segment are listed on top of them. The original motions in the 5 scenario segments are shown in Fig. 4.11 (b).

These motions are transformed according to the alignment between the skeleton of their scenarios and the new scenario. The motions after transformation are shown in Fig. 4.11 (c), and their *Collision Ratios* are listed below them. The phenomenon shown in Fig. 4.11 (a) (b) supports our previous hypothesis that the object's motions have close relationship with the change of scenario's skeleton and volume size. The motions always change greatly on the location where the scenario skeleton has great bend or sphere radius greatly fluctuate. The same conclusion can be found in Fig. 4.11 (c). The scenario segment most similar to the new scenario segment is the segment "59", the original motions in which can be directly transformed to the collision-free motions. It is noticed that the original motions in the scenario segment "23" can be transformed to generate new motions with very small *Collision Ratio*, even though it has a large dissimilarity to the new scenario segment in radius. One reason is that the segment "23" is very similar to the new segment in the bending of skeleton. Another reason is that all the spheres of the segment "23" are smaller than the new segment, which amplifies the dissimilarity in the radius but emphasizes the relationship between motions and skeleton. Because the new motions after transformation are collision-free, further verification is not necessary. The comparison of time spent in generating new motion between our method and RRT-connect algorithm is shown in Tab. 4.1. Our retrieval process has totally made 220 times comparison by DTW with each curve sampled into a sequence of 50 points. If replacing sequential search with search on R-tree, the distance calculation between  $(U_r, L_r)$  and MBR is much faster than DTW. Thus the retrieval in real time in a motion library which contains millions of scenario segments is possible. This result reveals the potential broad applications of our method in much further complex problems.

The results for the second selected scenario segment are demonstrated in Fig. 4.12. The search and motion transformation method are the same as the first one, which also supports our hypothesis. As in previous example, the 5 selected scenario segments with their original motions and transformed motions are shown in Fig. 4.12 (a) (b) (c) respectively. Both the scenario segments "88" and "61" are similar to the new scenario segment according to the radius change at the same level. However, the segment "61" is more similar to the new segment than the segment "88" according to skeleton's curvature, and thus its transformed motion has less *Collision Ratio* to the obstacles. Both the segments "61" and "65" are similar to the new segment according to skeleton's curvature at same level, one of which is in distance value 0.28 and the other is in distance value 0.31. However, compared to the segment "65", the segment "61" is much more similar to new segment according to radius change, and thus the original motions in the segment "61" is much more suitable to generate new motions with lower *Collision Ratio*. After transformation, the new motion trajectory still has some in-collision parts, so we sample some configurations in this new motion trajectory. 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 algorithm. This process (enclosed by red box in Fig. 4.12 (d)) compared with connection only by RRT-connect algorithm is shown in Fig. 4.12 (d). The result shows that the transformed motions by our method provide the local guidance to sampling based motion planners and thus greatly reduce the number of search trials in configuration space. The time spent by our method and RRT-connect algorithm is shown in tab. 4.2.

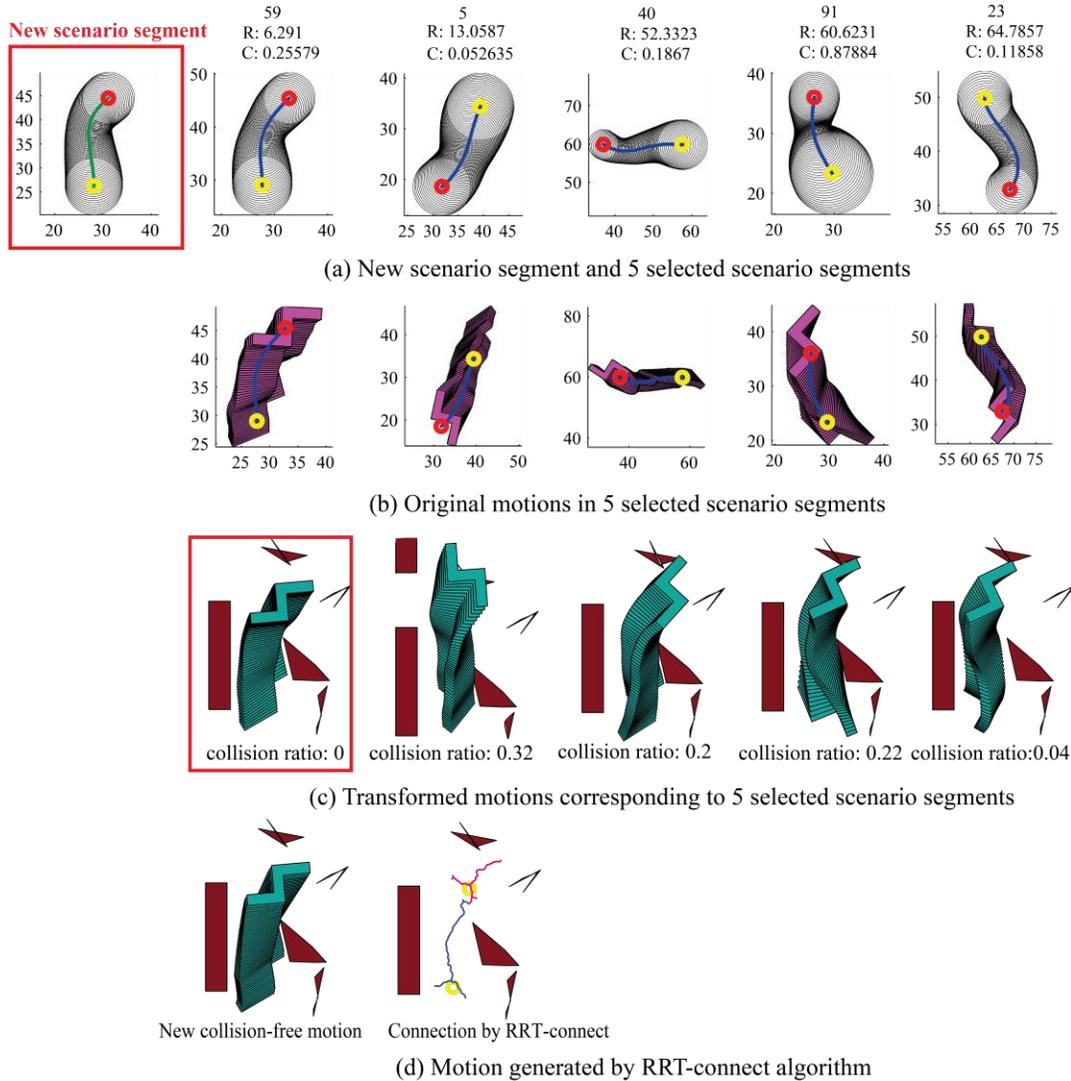


Figure 4.11. The motion generation for the first selected scenario segment and its comparison to RRT-connect

Table 4.1. Comparison between our method and RRT-connect on the motion generation in the first example

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

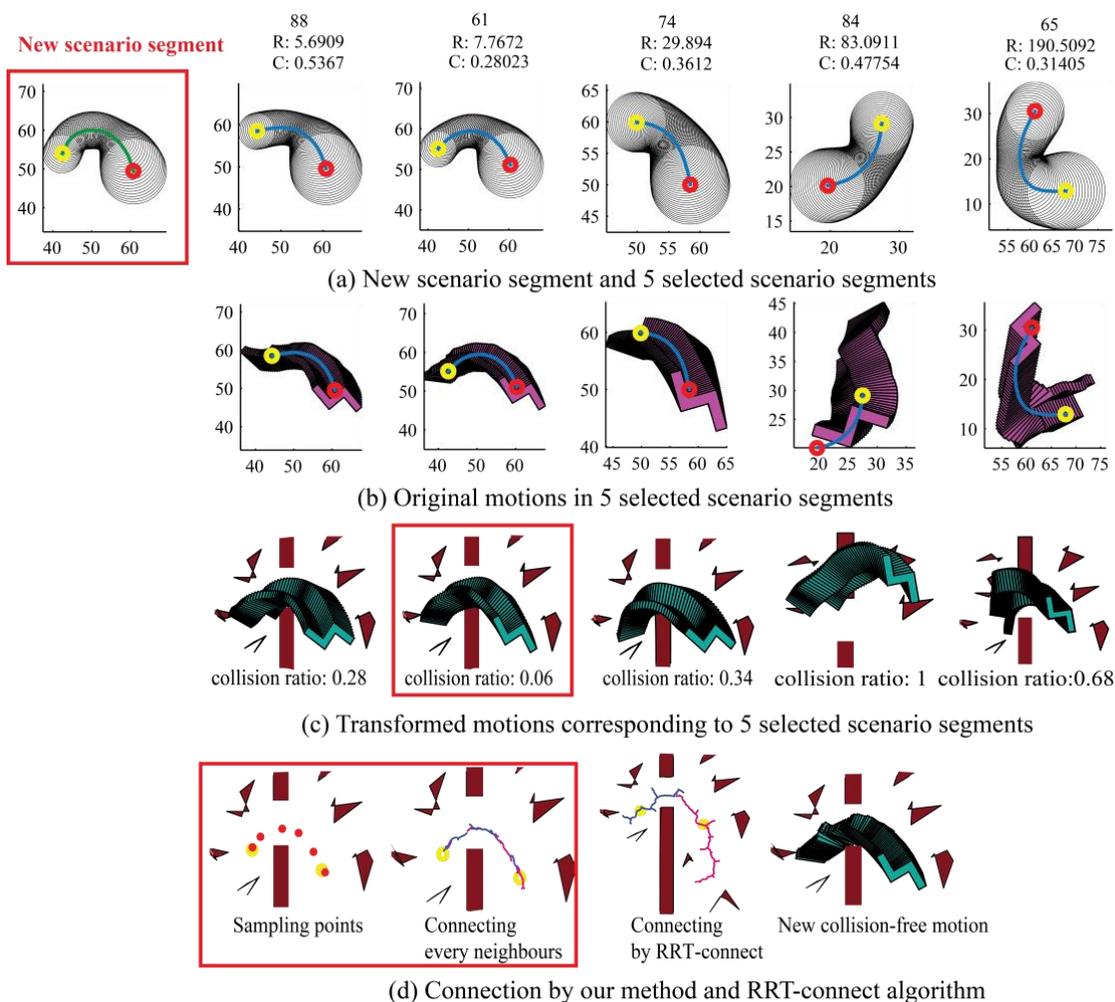


Figure 4.12. The motion generation for the second selected scenario segment and its comparison to RRT-connect

Table 4.2. Comparison between our method and RRT-connect on the motion generation in the second example

	$T_{total}(s)$	$T_{retrieval}(s)$	$T_{transform}(s)$	$Num_{retraction}$	$T_{retraction}(s)$	$T_{connection}(s)$
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

A complete planning result is shown in Fig. 4.13. In Fig. 4.13 (a), the search trace of RRT-connect algorithm is shown and followed by the smoothed motion. In Fig. 4.13 (b), the segmentation of the selected scenario is shown firstly. The scenario segments are marked by  $\{S_1, S_2, \dots, S_6\}$ . Then the

generated motion segments and the search trace are demonstrated. It is noticeable that, the retrieved motion in some scenario segments is collision-free, thus the further search is not necessary. The smoothed motion is shown in the end. The efficiency of our method is confirmed by the comparison of the search trace in Fig. 4.13 and the spent time in Tab. 4.3.

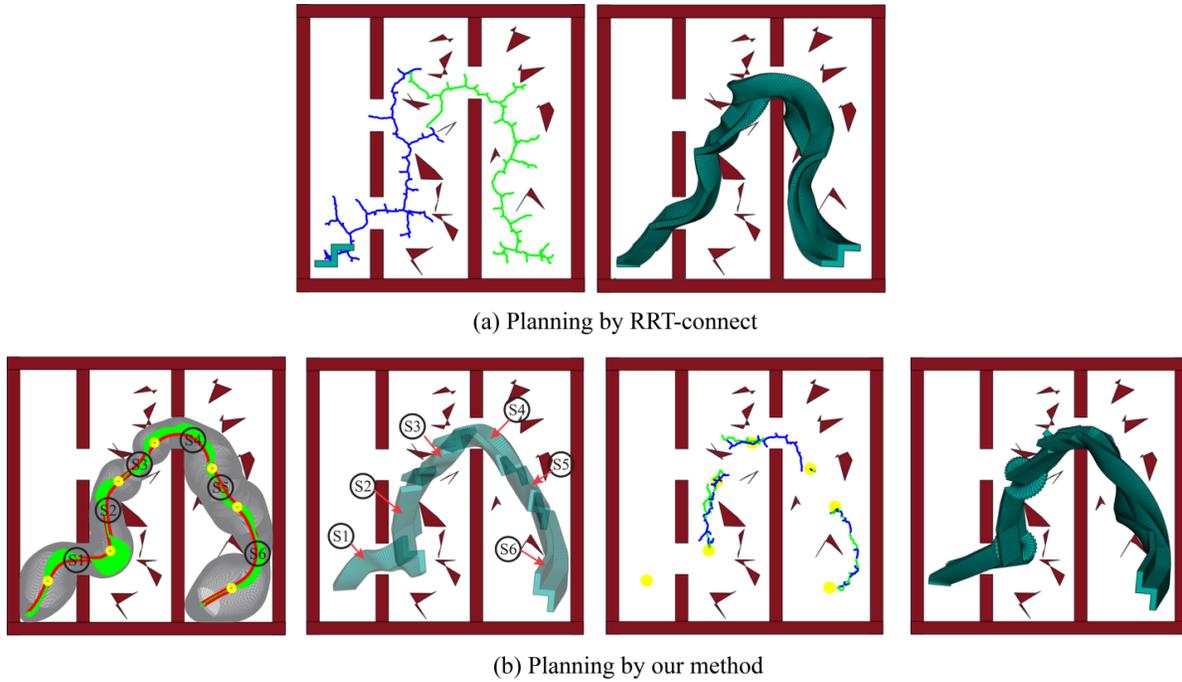


Figure 4.13. Comparison between our method and RRT-connect in a complete planning process

Table 4.3. Comparison between 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	N/A
		S2:	3.1607	0.2202	0.0739	0.1820	1	2.6846
		S3:	0.2963	0.2225	0.0738	N/A	N/A	N/A
		S4:	0.3141	0.2223	0.0918	N/A	N/A	N/A
		S5:	0.3172	0.2202	0.0970	N/A	N/A	N/A
		S6:	3.5896	0.2344	0.0979	0.1810	1	3.0763

### 4.3.2. Results in non-feasible scenarios

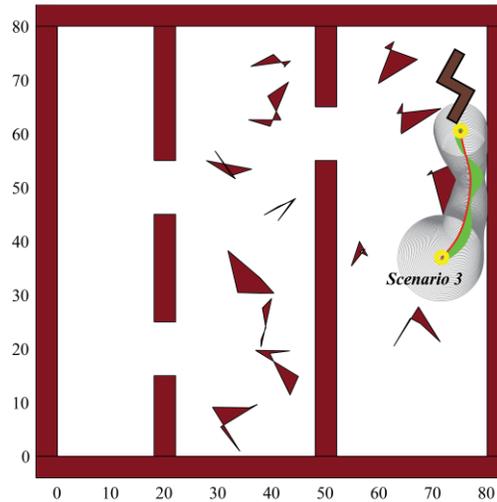


Figure 4.14. A non-feasible scenario segment selected during the interaction

In the environment of Fig. 4.10, the feasibility of some scenarios is not explicit, which means it is difficult for users to visually figure out whether the objective part could pass through the scenario or not. Such an example is shown in Fig. 4.14, the selected scenario seems possible for the part to pass through but actually it is not. A large proportion of narrow volume in the middle of the scenario causes the non-feasibility. And this is a practical problem in assembly design, because the assembly trajectory generated by automatic algorithm cannot guarantee human's preferences. It is necessary for designers to quickly check the feasibility of some interested scenarios. The initial search and motion transformation here are the same as the previous two cases. The search and transformation results are shown in Fig. 4.15 (a) (b) (c). The initial search is implemented in Successful Scenario segments. If the original motion of a selected scenario segment in search result can be transformed to a 'satisfactory' motion in new scenario, the repair phase is continued like the situation in the previous case, otherwise a new search is started in the Failure Scenario segments. The 'satisfactory' is defined by the condition that the minimum *Collision Ratio* of transformed motion is less than a threshold (0.5 here). In this case, the minimum *Collision Ratio* is 0.52, thus it is considered that none of the Successful Scenario segments in the motion library is useful for generating new motion. In order to detect if this scenario is essentially not feasible, the similar search is started in the Failure Scenario segments. The results are shown in Fig. 4.15 (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 part to pass through. The criterion 'similar' is defined by the similarity both in skeleton's curvature and sphere's radius which are less than 1 and 10 respectively. The Failure

Scenario segment “26” satisfies the ‘similar’ criterion, thus the new scenario is considered as non-feasible for the part to pass through. The further connection result (enclosed by red box shown in Fig. 4.15 (e)) verifies our “non-feasible” conclusion. The time spent for the detection of non-feasibility by our retrieval method is shown in the second line of Tab. 4.4. It is noticed that our method only spent 0.73 seconds to get the “non-feasible” conclusion, and the method in the second chapter spent 13 seconds. However, RRT-connect algorithm which has search priority in large volume space could not get the conclusion. If limiting the search of RRT-connect in the scenario segment, it spends much more time to confirm the non-feasibility of the scenario.

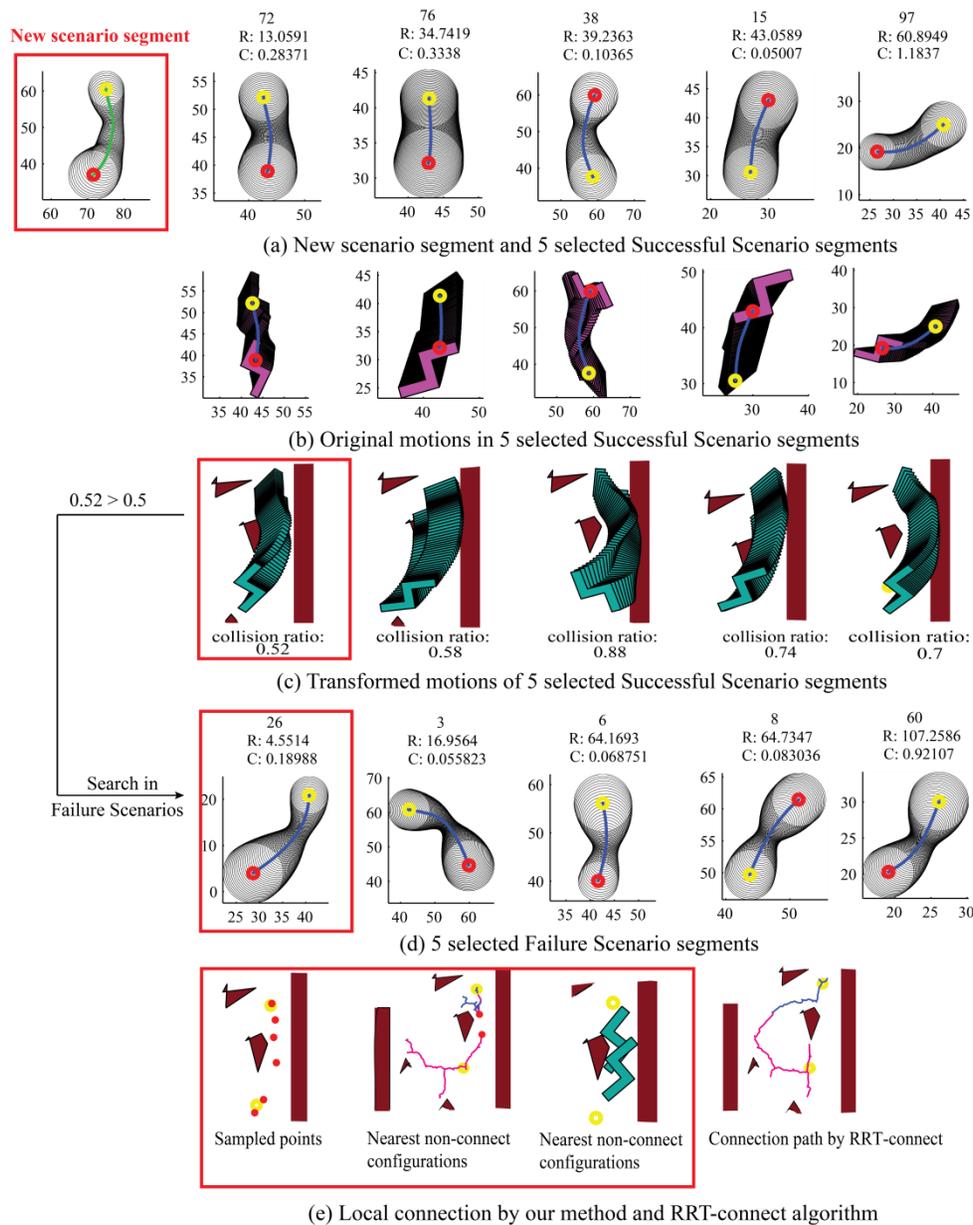


Figure 4.15. Feasibility detecting for the third selected scenario segment and its comparison to RRT-connect

Table 4.4. Comparison between our method and RRT-connect on the motion generation in the third example

	$T_{total}(s)$	$T_{retrieval}(s)$	$T_{transform}(s)$	$Num_{retraction}$	$T_{retraction}(s)$	$T_{connection}(s)$
RRT-connect	Fail	N/A	N/A	N/A	N/A	9.4459
Our detection	0.7347	0.3052	0.3181	0.1114	N/A	N/A
Further verification	13.6212	N/A	N/A	3	0.5090	13.1122

## 4.4. Extension

Until now, our motion learning method has been applied to the motion planning problem of the rigid free-flying object. In the following sections, the extensions to the motion planning of articulated object and the model with differential constraints are provided.

### 4.4.1. Extension to the motion planning of articulated object

The extension to the articulated object is essential because the accessibility of human body and robot arm are always considered in the product design. The situation for the articulated object is more complex. The complication concentrates on the design of motion transformation module. Recall our hypothesis that the topology of workspace has high correlation to the topology of  $\mathcal{CS}_{free}$ , and based on the hypothesis our method applies the motion segment in one scenario to the scenarios with similar topological structure. Although the universality of the hypothesis is intuitive and can be proved by empirical evidence, the motion transformation method in Section 4.2.3 is not universal.

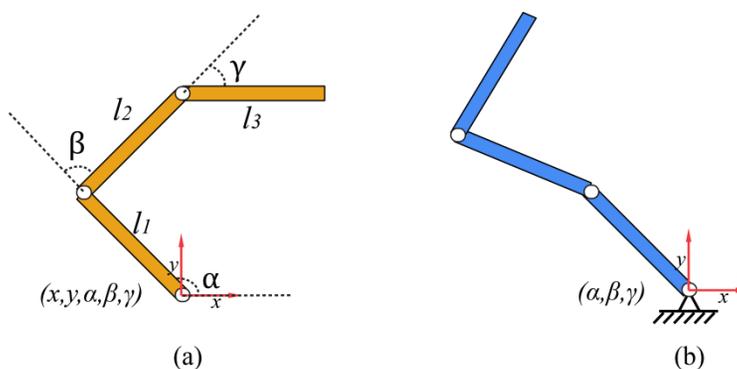


Figure 4.16. Two articulated models (The first is freely moveable in the workspace; the second is nailed to one point in the workspace)

Let us consider two ideal articulated objects in 2D workspace which are shown in Fig. 4.16. For both of the two objects, three bodies are attached to form a kinematic chain. The first object is

freely moveable with three revolute joints, and the second object is nailed to one point. The configuration of the first object can be represented as  $(x, y, \alpha, \beta, \gamma)$ , if  $\alpha, \beta, \gamma \in [-\pi, \pi]$  the configuration space is  $\mathbb{R}^2 \times \mathbb{T}^3$ . The configuration of the second object can be represented as  $(\alpha, \beta, \gamma)$ ,  $\alpha \in [0, \pi]$ , if  $\beta, \gamma \in [-\pi, \pi]$ , the configuration space is  $\mathbb{R} \times \mathbb{T}^2$ .

For the first object, if the action of two joint angles  $\beta, \gamma$  is not considered in the motion transformation phase, the transformation is implemented in  $\mathbb{R}^2 \times \mathbb{T}^1$ , which equals to the situation of the rigid body. Thus the motion segment in a scenario segment can be applied to its similar scenario segment through the translation and rotation in Section 4.2.3. In Fig. 4.17 (a), the scenario “2” is similar to the scenario “1”. The motion segment in scenario “1” is applied to the scenario “2” through the translation. It is obvious that, when not considering other obstacles, this application is tenable when scenario “2” has any various position and orientation in the 2D workspace.

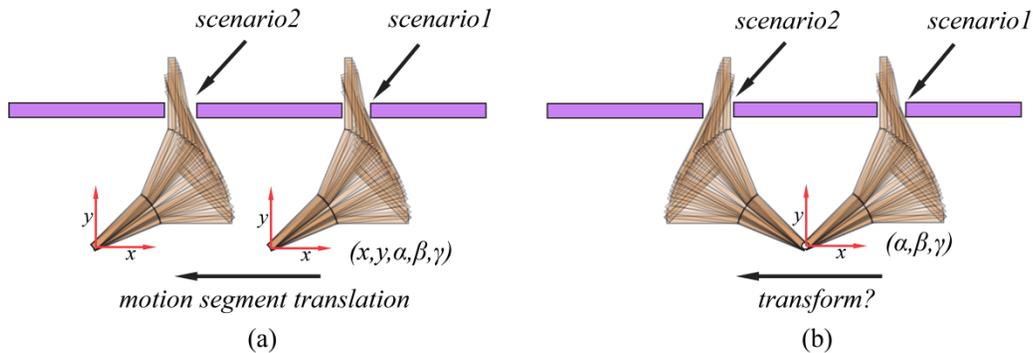


Figure 4.17. Examples of reusing motion segments of two articulated model

However, for the second articulated model, the translation and rotation cannot fulfill the requirement of motion transformation. In the Fig. 4.17 (b), the articulated model is nailed to a point, and the scenario “2” is axisymmetric to the scenario “1” about the Y axis which is across the nailed point. The motion segment in the scenario “1” can be applied to the scenario “2” through the reflection transformation. If the orientation of the scenario “2” is changed, some other affine transformations are needed. Here comes to the query that, if the topological structure of scenario still influences the topological structure of  $\mathcal{CS}_{free}$ ? In this following part, the empirical analysis is provided.

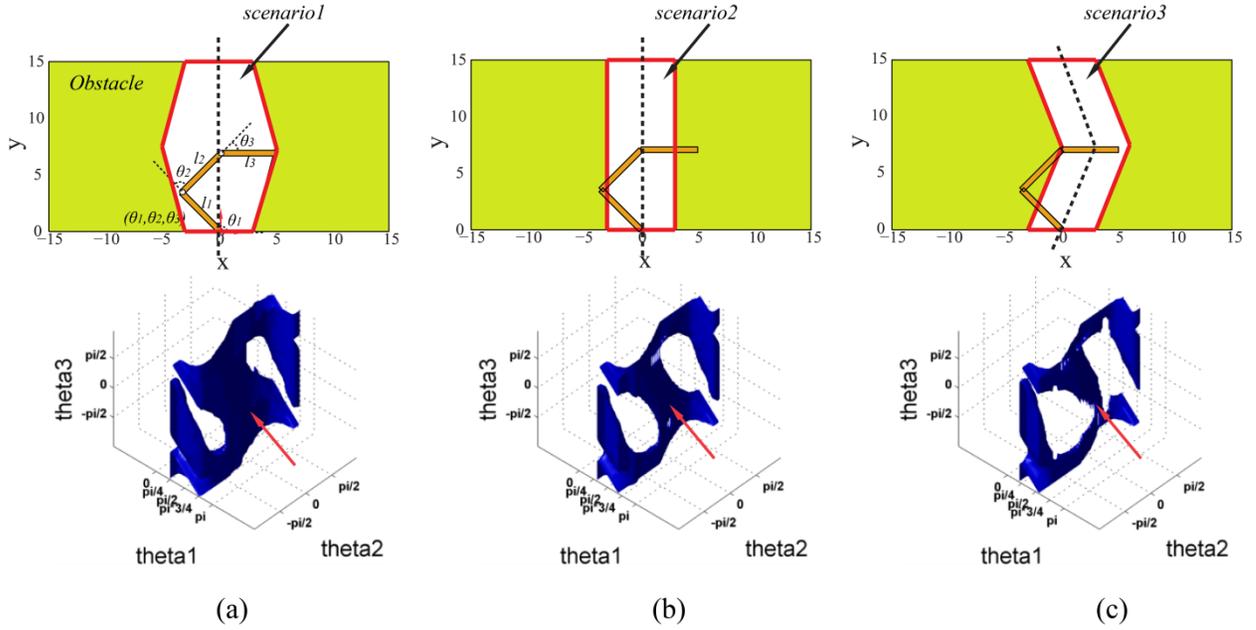


Figure 4.18. The comparison of the  $\mathcal{CS}_{free}$  of three different scenarios with  $\mathcal{CS} = \mathbb{R} \times \mathbb{T}^2$

For the second articulated object with one point nailed, three different scenarios are constructed as shown in the Fig. 4.18 top line. The topology of the three scenarios has been approximately described by the dashed lines, and the boundary of the scenarios has been marked by the red lines. The first and second scenarios have the same topology, but the first one is different to the second one with much larger volume in the middle part (Fig. 4.18 a-b). Although the volume of the third scenario is uniform, its topology is quite different to the previous two scenarios (Fig. 4.18 c). In this example, the configuration of the articulated object is represented as the  $(\theta_1, \theta_2, \theta_3)$ , in which  $\theta_1 \in [0, \pi]$  and  $\theta_2, \theta_3 \in [-\pi, \pi]$ . The collision-free configuration space  $\mathcal{CS}_{free}$  for each scenario is computed through checking the configurations uniformly sampled in the configuration space. The surfaces dividing the  $\mathcal{CS}_{free}$  and  $\mathcal{CS}_{obstacle}$  are indicated by the red arrows in the bottom line of Fig 4.18. The first and second scenarios are axisymmetric around the vertical axis across the origin, and their  $\mathcal{CS}_{free}$  shows the approximate point-symmetric around the point  $(\pi/2, 0, 0)$ . The reason is that, the length of the first body of the articulated object is set 5, the second joint which contributes the freedom  $\theta_2$  lies on the position  $(0, 5)$  when the articulated object is in the configuration  $(\pi/2, 0, 0)$ , and the shape of the first scenario is point-axisymmetric around the position  $(0, 7.5)$ . The large volume in the middle part of the first scenario provides more chances of actions than the second scenario, thus the first  $\mathcal{CS}_{free}$  is also enlarged, and the increase of  $\mathcal{CS}_{free}$  is also approximately point-symmetric. It is obvious that the topology of the  $\mathcal{CS}_{free}$  are the same for the first two scenarios. The change of the topology of the scenario will change the topology of  $\mathcal{CS}_{free}$ .

According to the previous analysis, it is not difficult to understand how the topology of the third scenario collapses the  $\mathcal{CS}_{free}$  and change the topology of the  $\mathcal{CS}_{free}$ , As shown in the Fig. 4.18 c.

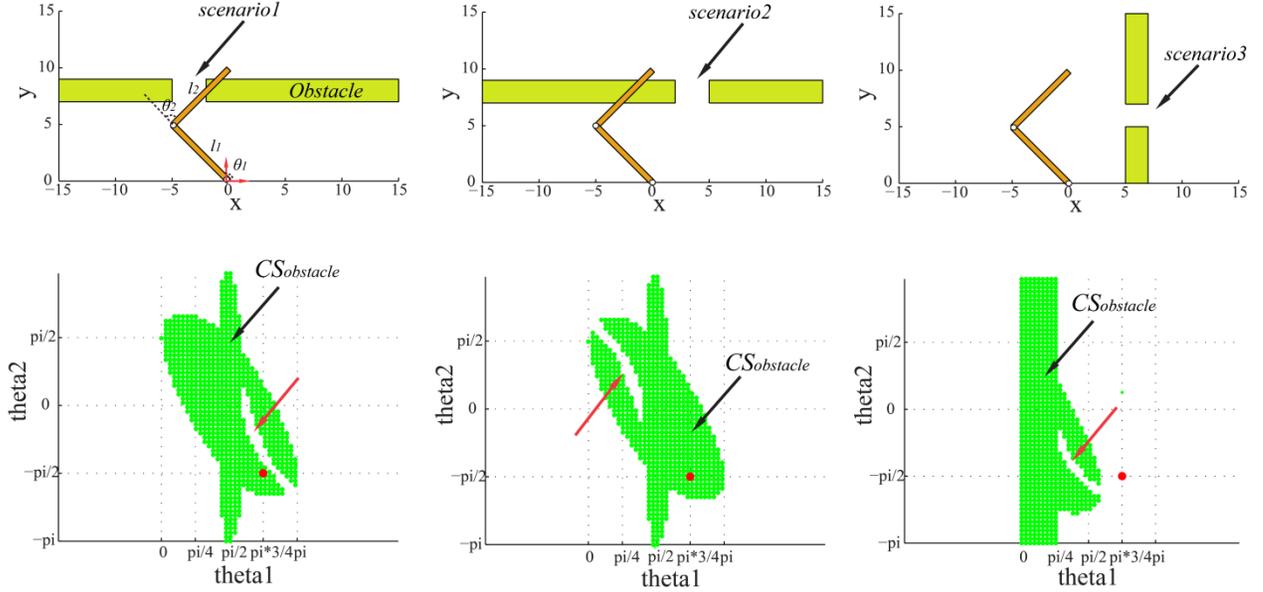


Figure 4.19. The comparison of the  $\mathcal{CS}_{free}$  of three different scenarios with  $\mathcal{CS} = \mathbb{R} \times \mathbb{T}$

In the last example, the three scenarios are in the same orientation. If our hypothesis is still tenable for the articulated object when the similar scenarios have the different positions and orientations? An example is shown in Fig. 4.19 to analysis this situation. In this example, the configuration of the articulated object is represented as the  $(\theta_1, \theta_2)$ , in which  $\theta_1 \in [0, \pi]$  and  $\theta_2 \in [-\pi, \pi]$ . All of the three constructed scenarios contain a small hole with the same shape (Fig. 4.19 top line). The configuration space of the articulated object is also computed through the uniform sampling. The in-collision configuration space  $\mathcal{CS}_{obstacle}$  is marked by the green color (Fig. 4.19 bottom line). In the  $\mathcal{CS}$  of the three cases, there is a narrow passage of  $\mathcal{CS}_{free}$  which invades into the  $\mathcal{CS}_{obstacle}$ . The narrow passage contains the configurations which indicate the actions of the articulated object inserting into the small hole. Although the small hole has the different positions and orientations in the three scenarios, the narrow passages of the  $\mathcal{CS}_{free}$  stay in the same topology in the three cases, because the small holes of the scenarios stay in the same topology.

From the examples above, it is not difficult to understand that the topology of the scenario imposes the decisive impact on the topology of the  $\mathcal{CS}_{free}$ . Thus our method is theoretically possible to be extended to high DOFs motion planning problem in which the configuration space is the product of multiple  $\mathbb{R}$  and  $\mathbb{T}$ . In spite of this, an exact mathematical proof is in debt and the general

transformation method of the motion segment is still not clear at present, which may involve the affine transformation in the configuration space.

#### 4.4.2. Extension to the differential motion planning

In the layout design or considering the logistics in the early design phase, the accessibility verification of the delivered freights and materials always takes into account the differential constraints introduced by the transporting tools (e.g. vehicle, aircraft...). In the previous chapters, the configuration of a rigid object moving in the 2D plane can be represented as  $q = (x, y, \theta)$ . While, considering the dynamics of the transporting tools, the state transition of the object should be subject to the velocity  $\dot{q} = (\dot{x}, \dot{y}, \dot{\theta})$  and acceleration  $\ddot{q} = (\ddot{x}, \ddot{y}, \ddot{\theta})$  constraints. The connection between the two configurations in the  $\mathcal{CS}$  is no longer a straight line but a curve whose curvature is determined by the differential constraints. One can imagine the scene of pulling a car to the curb within a crowded area. Since the wheels of the car cannot slide and be steered vertical to the curb, usually this task is difficult for a novice driver without a certain amount of training. Due to the introduction of the differential constraints, it is possible that the object cannot pass through a cluttered area even though the scenario is accessible for the object's geometrical size. This lead to the query, whether the motion segment in one scenario can be applied to the scenarios with the similar topological structure, when the motion of the object is under the differential constraints? If this hypothesis is still valid here, it makes sense that our efficient motion learning and retrieval strategy can be an enhancement to the differential motion planning solved by the other methods.

With the avoidance of sideslip and sharp steer, an intuitive experience shows that the velocity should be lower and the steering angle should be larger, when the bend is sharper and narrower. Thus, intuitively, the topological structure of the scenario strongly influences the velocity and the steering angle of the car. If the configuration transition equation of the simple car is  $\dot{q} = f(q, u)$  ( $u$  denotes the possible actions), it can be verified that  $u$  is determined by the velocity and the steering angle.

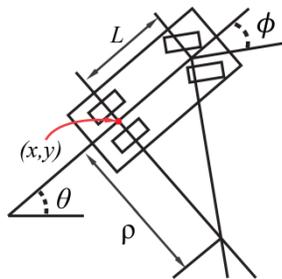


Figure 4.20. The model of a simple car

As shown in Fig. 4.20, if the configuration of the simple car is denoted by  $q = (x, y, \theta)$ , the steering angle is denoted by  $\phi$ , the body frame of the car places the origin at the center of the rear axle, the distance between the front and rear axles is represented as  $L$ , the  $\rho$  is the radius when the car has a circle motion, (LaValle 2006) indicated that the configuration transition equation for the simple car is:

$$\begin{aligned}\dot{x} &= s \cos(\theta) \\ \dot{y} &= s \sin(\theta) \\ \dot{\theta} &= \frac{s}{L} \tan(\phi)\end{aligned}\tag{3.1}$$

Thus the possible actions at any configuration of the simple car is determined by the velocity  $s$  and the steering angle  $\phi$ . It is reasonable to believe that, under the differential constraints, there also exists a close relationship between the topological structure of the scenario and the topological structure of the narrow passage. Usually, the steering angle of the car belongs to  $(-\phi_{max}, \phi_{max})$ , and  $\phi_{max} < \pi/2$ . If the velocity of the car is restricted as  $s \in \{0, 1\}$ , meaning that the car can only go forward in a constant speed or stop, then this car model is the Dubins Car see (Dubins 1957). In the following part, a small experiment is carried out to see how the topological structure of the scenario influences the path of the Dubins car.

In Fig. 4.21, the path of an Dubins car is planned by RRT algorithm through connecting the two configurations by an optimal path with the method in (Walker 2008). In this experiment, the Dubins car passes through three tightly constrained scenarios, each one of which has a bend of different curvature. The curvature of the bend descends from infinity in Fig. 4.21 (a) to zero in Fig. 4.21 (c). The RRT searching traces, the final paths and the scenarios with their skeletons are shown in the three columns in Fig. 4.21. The entrance and exit of the three scenario segments are shown in the last column. The width of the three scenario segments is designed very narrow so that almost the car can barely pass. This setting is especially on the purpose to eliminate the influence by the relationship between the volume of the scenario and the motion path. It is noteworthy that, in the three scenario segments, the path of the car obviously has a strong correlation with the topological structure of the scenario. In the outside of the scenario segment, the path of the car seems to have no obvious correlation with the scenario skeleton, because the influence by the enlarged volume, which increases the options of motion paths, cannot be ignored. This empirically proves that our previous hypothesis about the motion of the free-flying object is applicable to the object under the differential constraints. Thus it is reasonable to believe that our scenario matching and motion retrieval strategy can be applied to the motion planning under the differential constraints, even though it is still in debt with a theoretical proof.

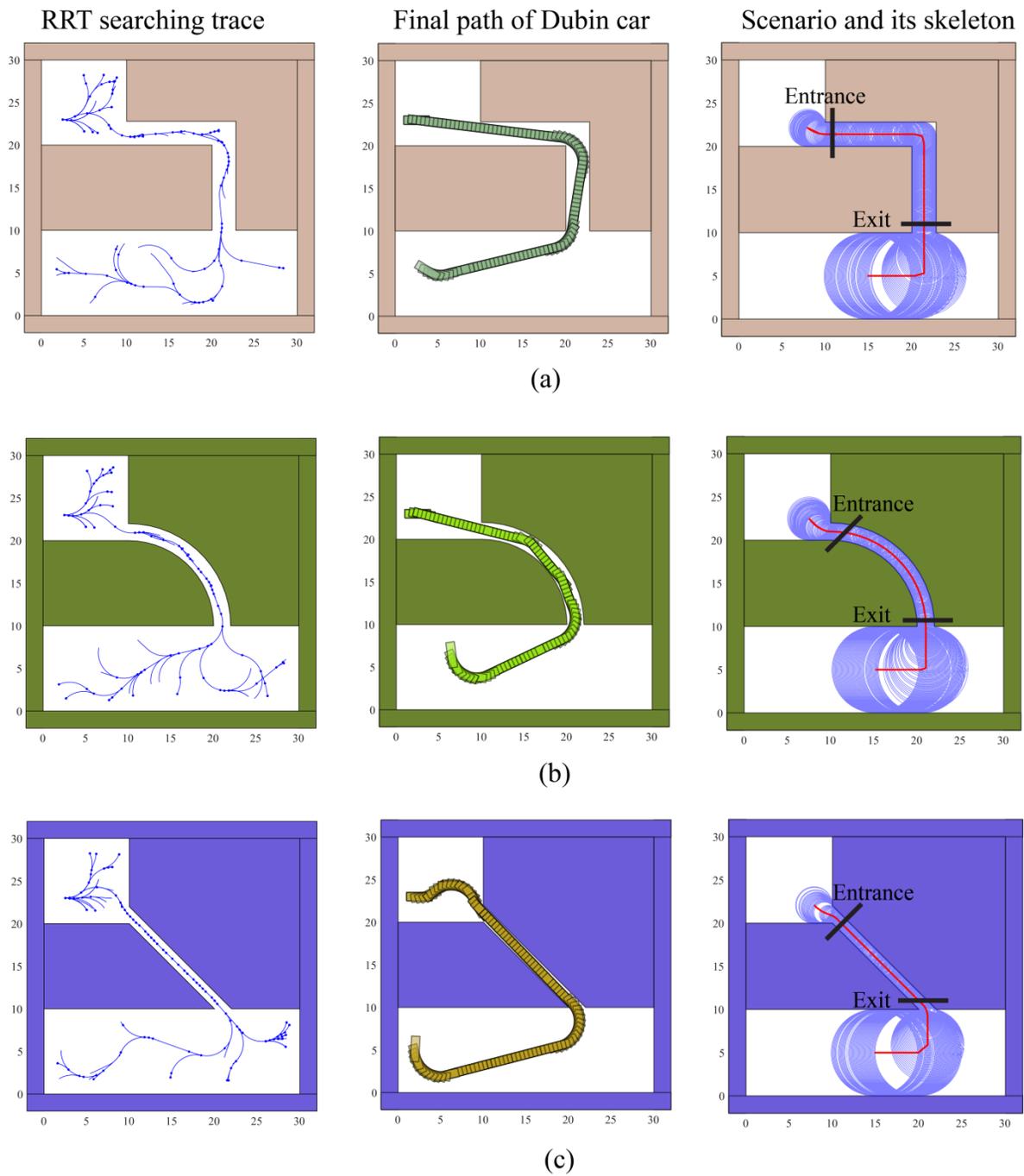


Figure 4.21. The planning results of the path of Dubin's car by RRT algorithm in three tightly constrained scenarios with different topological structures

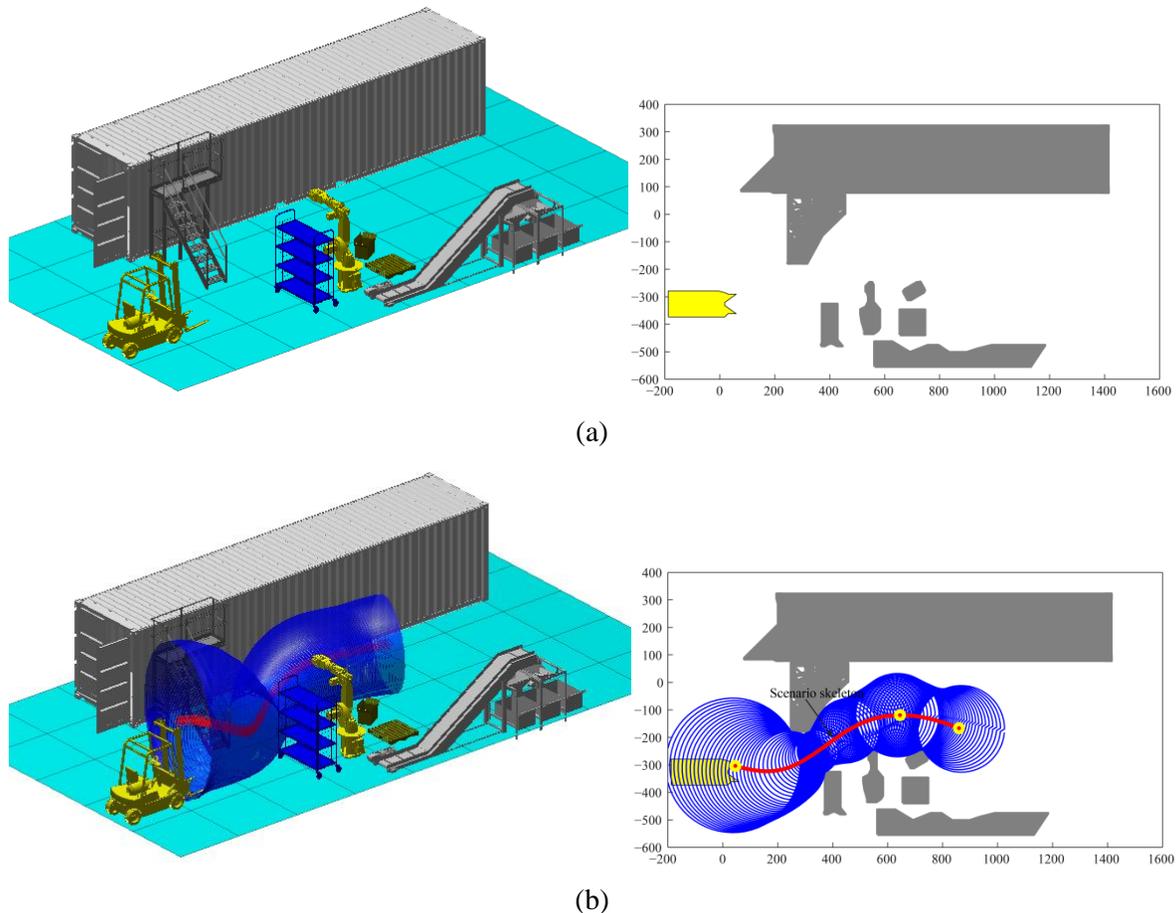


Figure 4.22. The Accessibility verification of a forklift in an industrial case with the construction of the scenario

Fig. 4.22 (a) demonstrates a possible scene in an industrial layout design example. In this example, the freight needs to be delivered by the forklift from the container to the range of the robot. When the accessibility of the plant layout is analyzed, the differential constraints of the forklift are taken into account. Since the motion planning of the forklift is considered frequently in the similar design processes, the construction of a method which can quickly get the conclusion of the accessibility based on the learned motion experience is very necessary. Similar to the method for the free-flying object, the scenario of the forklift and its skeleton can be easily obtained as in Fig. 4.22 (b). Our previous analysis inferred that the motion of objects under the differential constraints also has the correlation with the topological structure and the volume size of the scenario. Thus, for the scenario segment in Fig. 4.22 (b), if there is a scenario segment in motion library which has very similar scenario skeleton and the change of sphere radius, then the motion path in the memory scenario segment can be applied to this scenario segment. Since our scenario matching and retrieval method is very efficient, the real time search in a motion library, which stores enough number of scenario

segments to cover the practical scenarios, is possible. This also implies other possible applications of our method, such as the motion planning of the autonomous vehicles with the bounded curvature.

However, there is still some problems existing before our method can be successfully extended to the motion planning under the differential constraints. Similar to the situation of the articulated object, the problems focus on the motion transformation phase. In fact, our motion transformation carries out the translations, rotations and scaling to the motion path. The curvature is not invariant to the scaling transformations, thus there will be errors coming against the differential constraints when scaling the memory motion path. Supposing the curvature of the motion path is  $k$  and  $k = 1/R$  ( $R$  is the radius of the osculating circle), since  $\rho = L/\tan \phi$  in the previous case of the simple car, and  $R = \rho$ , thus  $k = \tan \phi / L$ . Noting that  $\phi \in (-\phi_{max}, \phi_{max})$ , and  $\phi_{max} < \pi/2$ , thus  $k \in (-\tan \phi_{max}/L, \tan \phi_{max}/L)$ . If the scaling stretches a motion path, the  $\|k\|$  will be reduced. If the scaling shrinks a motion path, the  $\|k\|$  will be increased, which may exceed the range. This property which resists the scaling transformation can also damage the motion repair method. Thus, the extension work is primarily on the motion transformation and repair phase.

A suggested change to our method is to fix the length of the scenario segment. The previous segmentation of the scenario is based on the volume change of the scenario, which can produce scenario segments of various lengths, and make the scaling of motion path indispensable. In order to avoid the scaling transformation, the scenario segments can be recorded and generated in a fixed length. This concept is similar to the method in (Knepper, Srinivasa et al. 2012). The major novelty is that, our method retrieves local motion paths from scenario sets rather than path sets. This change can improve the robustness of the motion retrieval, and greatly increase the chances that the retrieved motion segment can be directly applied to the new scenario segment. Due to our efficient scenario and motion retrieval, detecting in real time the accessibility of the possible tightly constrained scenario segments in front of the autonomous vehicles is feasible.

## 4.5. Conclusion

In this chapter, the interactive motion planning based on the learned experience has been proposed. The difficulty of the interaction between the user and the computer has been eased by permitting users to only control one point to select the scenarios in the workspace. Meanwhile, the user's selection is guided in real time by the automatic path planning of the controlled point. This interaction strategy can release users from some burdensome operations in the tightly constrained scenario, and break the bottleneck that the natural non-haptic devices cannot be used in the complex

tasks in which the manipulation of non-convex objects has to avoid collisions in a cluttered environment.

After scenario selection, the targeted scenario is split, and for each scenario segment, the most similar scenario segment is retrieved from the motion library, and the memory motion segment is transformed and repaired to fit the new scenario segment. Then, the new motion segments are stitched together. In the meantime, the inaccessibility of the new scenario segment can also be detected in a short time based on the information from the recorded inaccessible scenario segments. This provides a new perspective for the complete motion planning problem, which infers the inaccessible property of the scenario based on its similarity to the known inaccessible scenarios. 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. The possible extensions of our method to the articulated objects and the objects under differential constraints are discussed in the end of this chapter.



# 5 General Conclusions and perspectives

## Summary

In this thesis, a novel interactive motion planning framework with the learning ability has been constructed to address the difficulties of the accessibility verification in the product design process. The method created in the thesis can be used to check the accessibility of the rigid free-flying object in 2D and 3D Euclidean space, and some of the work can be extended to enhance the motion planning of the robot.

The first chapter firstly provides a comprehensive introduction to the accessibility verification problems in all the phases of the product lifecycle. Then the mathematical description to the accessibility problem has been offered along with the introduction to the difficulties for the automatic algorithms. Next, the limitations to the human manual manipulations in the virtual environment equipped with the interactive devices and in the real life have been analyzed. Finally, the motion learning techniques in the literature are compared.

In order to improve the performance of humans and algorithms on planning the motion in the tightly constrained local scenario, which is always accompanied with the narrow passage in the collision-free configuration space, a novel interactive motion planning method has been provided in the second chapter. In specific, an efficient motion retraction method is integrated into the interaction loop, which can not only ease the manipulations of humans, but also guide the local search of the automatic algorithms.

An original concept about the correlation between the topological structure of the scenario and the motion path in the configuration space is proposed in the beginning of the third chapter, along with the analysis to some empirical proofs. Based on the new concept, a novel motion learning method which explores the similarity of scenarios by the curve matching method has been created. An efficient management of the motion experience by a hierarchical structure is also created.

In the fourth chapter, based on the database of the learned motion segments, the manner of the interaction has been changed to the selection of scenarios rather than the manipulation of the object. Thus haptic and non-haptic devices show no remarkable differences in the interaction loop. The motion planning process has been changed to searching the similar scenario segments from the motion library to fit the targeted scenario segments and stitching the resulted motion segments together. The efficiency of our method has been highlighted in an example in 2D, in which the non-

convex object has to pass through the environment with the densely and randomly located non-convex obstacles.

## Future work

In the future, there are still a lot of problems and directions to explore based on the current work. Our retraction based interactive motion planning method depends highly on the efficient retraction method, which can retract the object from an in-collision configuration to a configuration in the free space or the contact space. The retraction is extremely important, primarily because it can not only produce a valid configuration in the difficult region, but also enhance the humans' understanding to their operations in the local region. An exact retraction process can be modeled as the problem of the *Generalized Penetration Depth*. The solutions to this problem have been provided by the efforts of several researchers. However, how to effectively integrate the results into an interaction loop and to serve the guidance to users has not been explored.

Our proposed hypothesis about the correlation between the topological structure of scenarios and the motion path in the configuration space is still an intuitive experience, but the mathematical proofs to whether or not it exists, and if so, it exists in which forms have not been answered in the literature. However, the importance of this hypothesis is, it runs in an opposite way to the sampling based algorithms, so that the correlation appears stronger if the passages in the free space become narrower, while the sampling based algorithms lose more power. And this correlation exists with different forms in almost all kinds of motion planning problems including the free-flying objects, the articulated objects, the objects under differential constraints and the deformable objects. Thus the theoretical proof to the correlation is worth much attention. And this correlation can provide a new dimension to describe the state of the motion, which can serve the more adaptive method using the machine learning algorithms.

Besides, the topological structure of the scenario is usually represented as a 1D curve, and the computation cost of the curve is much lower than the computation in the high DOFs configuration space, which makes the retrieval of the scenarios based on the similarity of its topological structure highly efficient. We have demonstrated in this thesis that the real time search in a motion library with millions of motion segments is possible. This can paved another road to the motion planning based on the cloud computing.

# Bibliography

Agrawal, R., C. Faloutsos and A. Swami (1993). Efficient similarity search in sequence databases. Foundations of Data Organization and Algorithms. D. Lomet, Springer Berlin Heidelberg. **730**: 69-84.

Amato, N. M., O. B. Bayazit, L. K. Dale, C. Jones and D. Vallejo (1998). OBPRM: an obstacle-based PRM for 3D workspaces. Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics: the algorithmic perspective, AK Peters, Ltd.

Argall, B. D., S. Chernova, M. Veloso and B. Browning (2009). "A survey of robot learning from demonstration." Robotics and autonomous systems **57**(5): 469-483.

Bayazit, O. B., G. Song and N. M. Amato (2001). "Enhancing randomized motion planners: Exploring with haptic hints." Autonomous Robots **10**(2): 163-174.

B é n a b è s, J. (2011). Optimisation int é g r é e et interactive pour l'agencement d'espace. PhD, Ecole Centrale de Nantes.

B é n a b è s, J., F. Bennis, E. Poirson and Y. Ravaut (2010). Accessibility in layout optimization. Proceedings of the EngOpt2010 Conference.

B é n a b è s, J., E. Poirson and F. Bennis (2013). "Integrated and interactive method for solving layout optimization problems." Expert Systems with Applications **40**(15): 5796-5803.

Berenson, D., P. Abbeel and K. Goldberg (2012). A robot path planning framework that learns from experience. Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE.

Berndt, D. J. and J. Clifford (1996). Finding patterns in time series: a dynamic programming approach. Advances in knowledge discovery and data mining. M. F. Usama, P.-S. Gregory, S. Padhraic and U. Ramasamy, American Association for Artificial Intelligence: 229-248.

Bollobás, B., G. Das, D. Gunopulos and H. Mannila (1997). Time-series similarity problems and well-separated geometric sets. Proceedings of the thirteenth annual symposium on Computational geometry. Nice, France, ACM: 454-456.

Boor, V., M. H. Overmars and A. F. van der Stappen (1999). The gaussian sampling strategy for probabilistic roadmap planners. Proceedings of IEEE International Conference on Robotics and Automation, IEEE.

Bouzit, M., G. Burdea, G. Popescu and R. Boian (2002). "The Rutgers Master II-new design force-feedback glove." Mechatronics, IEEE/ASME Transactions on **7**(2): 256-263.

- Branicky, M. S., R. A. Knepper and J. J. Kuffner (2008). Path and trajectory diversity: Theory and algorithms. Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE.
- Burdea, G. C. (1999). Keynote address: haptics feedback for virtual reality. Proceedings of international workshop on virtual prototyping. Laval, France.
- Burns, B. and O. Brock (2005). Sampling-based motion planning using predictive models. Proceedings of the IEEE International Conference on Robotics and Automation, IEEE.
- Burns, B. and O. Brock (2005). Toward Optimal Configuration Space Sampling. Robotics: Science and Systems, Citeseer.
- Chang, H. and T.-Y. Li (1995). Assembly maintainability study with motion planning. Proceedings of IEEE International Conference on Robotics and Automation, IEEE.
- Chang, M.-C. and B. B. Kimia (2011). "Measuring 3D shape similarity by graph-based matching of the medial scaffolds." Computer Vision and Image Understanding **115**(5): 707-720.
- Chen, L., M. T. zsu and V. Oria (2005). Robust and fast similarity search for moving object trajectories. Proceedings of the 2005 ACM SIGMOD international conference on Management of data. Baltimore, Maryland, ACM: 491-502.
- Cornea, N. D., D. Silver and P. Min (2007). "Curve-skeleton properties, applications, and algorithms." IEEE Transactions on Visualization and Computer Graphics **13**(3): 530-548.
- Cui, M., J. Femiani, J. Hu, P. Wonka and A. Razdan (2009). "Curve matching for open 2D curves." Pattern Recognition Letters **30**(1): 1-10.
- Denny, J. and N. M. Amato (2013). Toggle PRM: A Coordinated Mapping of C-Free and C-Obstacle in Arbitrary Dimension. Algorithmic Foundations of Robotics X. E. Frazzoli, T. Lozano-Perez, N. Roy and D. Rus, Springer Berlin Heidelberg. **86**: 297-312.
- Dragan, A., G. Gordon and S. Srinivasa (2011). Learning from experience in manipulation planning: Setting the right goals. Proceedings of the International Symposium on Robotics Research (ISRR).
- Dubins, L. E. (1957). "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents." American Journal of mathematics: 497-516.
- Erbes, R. (2013). Efficient parallel proximity queries and an application to highly complex motion planning problems with many narrow passages, Universit ätsbibliothek Mainz.
- Ferre, E. and J.-P. Laumond (2004). An iterative diffusion algorithm for part disassembly. IEEE International Conference on Robotics and Automation, IEEE.

Ferré E., J.-P. Laumond, G. Arechavaleta and C. Estevès (2005). Progresses in assembly path planning. International Conference on Product Lifecycle Management: 373-382.

Fillatreau, P., J. Y. Fourquet, R. Le Bolloc'h, S. Cailhol, A. Datas and B. Puel (2013). "Using virtual reality and 3D industrial numerical models for immersive interactive checklists." Computers in Industry **64**(9): 1253-1262.

Finney, S., L. P. Kaelbling and T. Lozano-Pérez (2007). Predicting Partial Paths from Planning Problem Parameters. Robotics: Science and Systems.

Flavigné D. and M. Taïk (2011). Interactive locomotion animation using path planning. IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), IEEE.

Fonte, D. (2011). Motion Planning for a Rhombic-Like Vehicle Operating in ITER Scenarios, Ph. D. dissertation, Instituto Superior Técnico.

Garber, M. and M. C. Lin (2002). Constraint-based motion planning for virtual prototyping. Proceedings of the seventh ACM symposium on Solid modeling and applications, ACM.

Geng, J., D. Zhou, C. Lv and Z. Wang (2013). "A modeling approach for maintenance safety evaluation in a virtual maintenance environment." Computer-Aided Design **45**(5): 937-949.

Geraerts, R. and M. Overmars (2004). A Comparative Study of Probabilistic Roadmap Planners. Algorithmic Foundations of Robotics V. J.-D. Boissonnat, J. Burdick, K. Goldberg and S. Hutchinson, Springer Berlin Heidelberg. **7**: 43-58.

Geraerts, R. and M. H. Overmars (2007). The Corridor Map Method: real-time high-quality path planning. International Conference on Robotics and Automation, ICRA.

Gomes de Sá A. and G. Zachmann (1999). "Virtual reality as a tool for verification of assembly and maintenance processes." Computers & Graphics **23**(3): 389-403.

Gope, C., N. Kehtarnavaz, G. Hillman and B. Würsig (2005). "An affine invariant curve matching method for photo-identification of marine mammals." Pattern Recognition **38**(1): 125-132.

Gottschalk, S., M. C. Lin and D. Manocha (1996). OBBTree: A hierarchical structure for rapid interference detection. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM.

Guttman, A. (1984). R-trees: a dynamic index structure for spatial searching. Proceedings of the 1984 ACM SIGMOD international conference on Management of data. Boston, Massachusetts, ACM: 47-57.

Guzmán, J. L., M. Berenguel, F. Rodríguez and S. Dormido (2008). "An interactive tool for mobile robot motion planning." Robotics and Autonomous Systems **56**(5): 396-409.

- Hauser, K., T. Bretl, K. Harada and J.-C. Latombe (2008). Using motion primitives in probabilistic sample-based planning for humanoid Robots. Algorithmic Foundation of Robotics VII. S. Akella, N. Amato, W. Huang and B. Mishra, Springer Berlin Heidelberg. **47**: 507-522.
- He, X. and Y. Chen (2009). "Haptic-aided robot path planning based on virtual tele-operation." Robotics and Computer-Integrated Manufacturing **25**(4-5): 792-803.
- Hermansson, T., R. Bohlin, J. S. Carlson and R. Söderberg (2013). "Automatic assembly path planning for wiring harness installations." Journal of Manufacturing Systems **32**(3): 417-422.
- Howard, T. M., M. Pivtoraiko, R. A. Knepper and A. Kelly (2014). "Model-Predictive Motion Planning." IEEE ROBOTICS & AUTOMATION MAGAZINE.
- Hsin-Yi, Y., S. Thomas, D. Eppstein and N. M. Amato (2012). UOBPRM: A uniformly distributed obstacle-based PRM. International Conference on Intelligent Robots and Systems (IROS).
- Hsu, D., T. Jiang, J. Reif and Z. Sun (2003). The bridge test for sampling narrow passages with probabilistic roadmap planners. Proceedings of IEEE International Conference on Robotics and Automation, ICRA'03, IEEE.
- Hsu, D., L. E. Kavraki, J.-C. Latombe, R. Motwani and S. Sorkin (1998). On finding narrow passages with probabilistic roadmap planners. Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics.
- Hsu, D., G. Sánchez-Ante, H.-I. Cheng and J.-C. Latombe (2006). Multi-level free-space dilation for sampling narrow passages in PRM planning. Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, IEEE.
- Hsu, D., G. Sánchez-Ante and Z. Sun (2005). Hybrid PRM sampling with a cost-sensitive adaptive strategy. Proceedings of the IEEE International Conference on Robotics and Automation, IEEE.
- Hsu, H.-Y. and G. C. I. Lin (2002). "Quantitative measurement of component accessibility and product assemblability for design for assembly application." Robotics and Computer-Integrated Manufacturing **18**(1): 13-27.
- Hwang, Y. K., K. R. Cho, L. Sooyong, S. M. Park and K. Sungchul (1997). Human computer cooperation in interactive motion planning. Proceedings of 8th International Conference on Advanced Robotics, ICAR '97, 1997.
- Jacquenot, G. (2008, Updated 16 Dec 2008). "Minimum distance between two polygons." Matlab File Exchange, from <http://www.mathworks.fr/matlabcentral/fileexchange/22444-minimum-distance-between-two-polygons>.

Jacquenot, G. (2010). Méthode générique pour l'optimisation d'agencement géométrique et fonctionnel, Ecole centrale de Nantes-ECN.

Jang, H.-Y., H. Moradi, P. Le Minh, S. Lee and J. Han (2008). "Visibility-based spatial reasoning for object manipulation in cluttered environments." Computer-Aided Design **40**(4): 422-438.

Jaramillo, C. E. E. (2007). Motion planning: from digital actors to humanoid robots. PhD, Institut National Polytechnique de Toulouse-INPT.

Jayaram, S., U. Jayaram, Y. J. Kim, C. DeChenne, K. W. Lyons, C. Palmer and T. Mitsui (2007). "Industry case studies in the use of immersive virtual assembly." Virtual Reality **11**(4): 217-228.

Je, C., M. Tang, Y. Lee, M. Lee and Y. J. Kim (2012). "Polydepth: Real-time penetration depth computation using iterative contact-space projection." ACM Transactions on Graphics (TOG) **31**(1): 5.

Jetchev, N. and M. Toussaint (2013). "Fast motion planning from experience: trajectory prediction for speeding up movement generation." Autonomous Robots **34**(1-2): 111-127.

Jetchev, N. N. (2012). Learning representations from motion trajectories: Analysis and applications to robot planning and control. PhD, Freie Universität Berlin.

Ji, X. and J. Xiao (2001). "Planning motions compliant to complex contact states." The International Journal of Robotics Research **20**(6): 446-465.

Jiang, W., K. Xu, Z.-Q. Cheng, R. R. Martin and G. Dang (2013). "Curve skeleton extraction by coupled graph contraction and surface clustering." Graphical Models **75**(3): 137-148.

Kallmann, M., R. Bargmann and M. Mataric (2004). Planning the sequencing of movement primitives. proceedings of the international conference on simulation of adaptive behavior (SAB).

Kavraki, L. E., P. Svestka, J.-C. Latombe and M. H. Overmars (1996). "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." IEEE Transactions on Robotics and Automation **12**(4): 566-580.

Keller, D., L. Doceul, F. Ferlay, G. Jiolat, J. Cordier, I. Kuehn, B. Manfreo and J. Reich (2013). "ITER design, integration and assembly studies assisted by virtual reality." Fusion Engineering and Design **88**(9): 1951-1954.

Keogh, E. and C. A. Ratanamahatana (2005). "Exact indexing of dynamic time warping." Knowledge and Information Systems **7**(3): 358-386.

Khalid, S. (2012). "Incremental indexing and retrieval mechanism for scalable and robust shape matching." Multimedia systems **18**(4): 319-336.

- Kim, Y. J., M. A. Otaduy, M. C. Lin and D. Manocha (2003). Fast penetration depth estimation using rasterization hardware and hierarchical refinement. Proceedings of the nineteenth annual symposium on Computational geometry, ACM.
- Kin - Chung Au, O., C. L. Tai, D. Cohen - Or, Y. Zheng and H. Fu (2010). Electors voting for fast automatic shape correspondence. Computer Graphics Forum, Wiley Online Library.
- Knepper, R. A. and M. T. Mason (2009). Empirical sampling of path sets for local area motion planning. Experimental Robotics, Springer.
- Knepper, R. A., S. S. Srinivasa and M. T. Mason (2012). "Toward a deeper understanding of motion alternatives via an equivalence relation on local paths." The International Journal of Robotics Research **31**(2): 167-186.
- Kroon, D.-J. (2011, Updated 17 Oct 2011). "2D Line curvature and normals." Matlab File Exchange, from <http://www.mathworks.fr/matlabcentral/fileexchange/32696-2d-line-curvature-and-normals>.
- Kuffner, J. J. and S. M. LaValle (2000). RRT-connect: An efficient approach to single-query path planning. Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, IEEE.
- Kustra, J., A. Jalba and A. Telea (2013). Probabilistic view-based 3D curve skeleton computation on the GPU. Proceedings of 8th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. Barcelona, Spain. **2**: 237-246.
- Ladeveze, N., J.-Y. Fourquet, B. Puel and M. Taix (2009). Haptic assembly and disassembly task assistance using interactive path planning. IEEE Virtual Reality Conference. VR'09, 2009.
- Lamiroux, F., D. Bonnafous and C. Van Geem (2003). Path Optimization for Nonholonomic Systems: Application to Reactive Obstacle Avoidance and Path Planning. Control Problems in Robotics. A. Bicchi, D. Prattichizzo and H. Christensen, Springer Berlin Heidelberg. **4**: 1-18.
- Lamiroux, F., J.-P. Laumond, C. Van Geem, D. Boutonnet and G. Raust (2005). "Trailer truck trajectory optimization: the transportation of components for the Airbus A380." IEEE Robotics & Automation Magazine **12**(1): 14-21.
- Larsen, E., S. Gottschalk, M. C. Lin and D. Manocha (1999). Fast proximity queries with swept sphere volumes, Technical Report TR99-018, Department of Computer Science, University of North Carolina.
- Latombe, J.-C. (1991). Configuration Space of a Rigid Object. Robot Motion Planning, Springer US. **124**: 58-104.

Latombe, J.-C. (1999). "Motion planning: A journey of robots, molecules, digital actors, and other artifacts." The International Journal of Robotics Research **18**(11): 1119-1128.

Laumond, J.-P. (2006). "Motion planning for PLM: state of the art and perspectives." International Journal of Product Lifecycle Management **1**(2): 129-142.

Lavalle, S., J. Kuffner and Jr (2000). Rapidly-Exploring Random Trees: Progress and Prospects. Algorithmic and Computational Robotics: New Directions.

Lavalle, S. M. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning. Computer Science Department, Iowa State University.

LaValle, S. M. (2006). Planning algorithms, Cambridge university press.

Lee, J., O. Kwon, L. Zhang and S.-e. Yoon (2012). Sr-rrt: Selective retraction-based rrt planner. Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE.

Leu, M. C., H. A. ElMaraghy, A. Y. Nee, S. K. Ong, M. Lanzetta, M. Putz, W. Zhu and A. Bernard (2013). "CAD model based virtual assembly simulation, planning and training." CIRP Annals-Manufacturing Technology **62**(2): 799-822.

Liangjun, Z. and D. Manocha (2008). An efficient retraction-based RRT planner. International Conference on Robotics and Automation, ICRA.

Lien, J.-M. (2008). "Hybrid motion planning using Minkowski sums." Proceedings of robotics: science and systems IV.

Lien, J.-M., J. Keyser and N. M. Amato (2006). Simultaneous shape decomposition and skeletonization. Proceedings of the 2006 ACM symposium on Solid and physical modeling. Cardiff, Wales, United Kingdom, ACM: 219-228.

Lien, J.-M. and Y. Lu (2009). Planning motion in environments with similar obstacles. Robotics: Science and Systems, Citeseer.

Lien, J.-M., S. L. Thomas and N. M. Amato (2003). A general framework for sampling on the medial axis of the free space. Proceedings of IEEE International Conference on Robotics and Automation ICRA'03, IEEE.

Lozano-Perez, T. (1983). "Spatial planning: A configuration space approach." IEEE Transactions on Computers **100**(2): 108-120.

Lu, W. and J.-F. Petiot (2014). "Affective design of products using an audio-based protocol: Application to eyeglass frame." International Journal of Industrial Ergonomics **44**(3): 383-394.

Ma, J., S. Bae and S. Choi (2012). "3D medial axis point approximation using nearest neighbors and the normal field." The Visual Computer **28**(1): 7-19.

Manolopoulos, Y., A. Nanopoulos, A. Papadopoulos and Y. Theodoridis (2006). Introduction. R-Trees: Theory and Applications, Springer London: 3-13.

Maropoulos, P. G. and D. Ceglarek (2010). "Design verification and validation in product lifecycle." CIRP Annals - Manufacturing Technology **59**(2): 740-759.

Michalos, G., S. Makris, N. Papakostas, D. Mourtzis and G. Chryssolouris (2010). "Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach." CIRP Journal of Manufacturing Science and Technology **2**(2): 81-91.

Mitchell, T. M. (1997). Machine Learning, McGraw-Hill, Inc.

Morales, A., L. Tapia, R. Pearce, S. Rodriguez and N. M. Amato (2005). C-space subdivision and integration in feature-sensitive motion planning. Proceedings of IEEE International Conference on Robotics and Automation, IEEE.

Morales, M., L. Tapia, R. Pearce, S. Rodriguez and N. M. Amato (2005). A machine learning approach for feature-sensitive motion planning. Algorithmic Foundations of Robotics VI, Springer: 361-376.

Morato, C., K. N. Kaipa and S. K. Gupta (2013). "Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters." Computer-Aided Design **45**(11): 1349-1364.

Müller, M. (2007). Information retrieval for music and motion, Springer.

Nelaturi, S. (2011). Configuration modeling. PhD, UNIVERSITY OF WISCONSIN.

Nelaturi, S., M. Lysenko and V. Shapiro (2012). "Rapid Mapping and Exploration of Configuration Space." Journal of Computing and Information Science in Engineering **12**(2): 021007-021007.

Nelaturi, S. and V. Shapiro (2013). "Solving inverse configuration space problems by adaptive sampling." Computer-Aided Design **45**(2): 373-382.

Nguyen, V.-H., F. Merienne and J.-L. Martinez (2010). An Efficient Approach for Human Motion Data Mining Based on Curves Matching. Computer Vision and Graphics. L. Bolc, R. Tadeusiewicz, L. Chmielewski and K. Wojciechowski, Springer Berlin Heidelberg. **6374**: 163-184.

Pan, J., S. Chitta and D. Manocha (2013). Faster sample-based motion planning using instance-based learning. Algorithmic Foundations of Robotics X. E. Frazzoli, T. Lozano-Perez, N. Roy and D. Rus, Springer Berlin Heidelberg. **86**: 381-396.

Pan, J., L. Zhang and D. Manocha (2010). Retraction-based RRT planner for articulated models. Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE.

- Pan, J., X. Zhang and D. Manocha (2013). "Efficient penetration depth approximation using active learning." ACM TRANSACTIONS ON GRAPHICS **32**(6).
- Park, H. S., C. H. Choi, S. H. Kim, B. S. Park, K. H. Kim and H. D. Kim (2011). "Deployment analysis and remote accessibility verification for a maintenance task in a PRIDE digital mock-up." Annals of Nuclear Energy **38**(4): 767-774.
- Pastor, P., H. Hoffmann, T. Asfour and S. Schaal (2009). Learning and generalization of motor skills by learning from demonstration. International Conference on Robotics and Automation, ICRA'09, IEEE.
- Patil, S. and R. Alterovitz (2010). Interactive motion planning for steerable needles in 3D environments with obstacles. Biomedical Robotics and Biomechanics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on, IEEE.
- Pelekis, N., I. Kopanakis, G. Marketos, I. Ntoutsis, G. Andrienko and Y. Theodoridis (2007). Similarity Search in Trajectory Databases. Proceedings of the 14th International Symposium on Temporal Representation and Reasoning, IEEE Computer Society: 129-140.
- Peng Tang and Jing Xiao (2008). "Automatic generation of high-level contact state space between 3D curved objects." The International Journal of Robotics Research **27**(7): 832-854.
- Poirson, E., J.-F. Petiot, L. Boivin and D. Blumenthal (2013). "Eliciting user perceptions using assessment tests based on an interactive genetic algorithm." Journal of Mechanical Design **135**(3): 031004-031004.
- Popescu, V., G. Burdea and M. Bouzit (1999). Virtual reality simulation modeling for a haptic glove. Proceedings of Computer Animation, IEEE.
- Rajan, V. N., K. Sivasubramanian and J. E. Fernandez (1999). "Accessibility and ergonomic analysis of assembly product and jig designs." International Journal of Industrial Ergonomics **23**(5-6): 473-487.
- Ratliff, N., M. Zucker, J. A. Bagnell and S. Srinivasa (2009). CHOMP: Gradient optimization techniques for efficient motion planning. Proceedings of IEEE International Conference on Robotics and Automation, ICRA, IEEE.
- Redon, S. and M. C. Lin (2005). Practical local planning in the contact space. Proceedings of International Conference on Robotics and Automation, ICRA, IEEE.
- Rodriguez, S., S. Thomas, R. Pearce and N. M. Amato (2008). Resampl: A region-sensitive adaptive motion planner. Algorithmic Foundation of Robotics VII, Springer: 285-300.
- Rodriguez, S., T. Xinyu, L. Jyh-Ming and N. M. Amato (2006). An obstacle-based rapidly-exploring random tree. Proceedings of International Conference on Robotics and Automation, ICRA.

- Rosell, J., C. Vázquez, A. Pérez and P. Iñiguez (2008). "Motion planning for haptic guidance." Journal of Intelligent and Robotic Systems **53**(3): 223-245.
- Saha, M., J.-C. Latombe, Y.-C. Chang and F. Prinz (2005). "Finding narrow passages with probabilistic roadmaps: The small-step retraction method." Autonomous robots **19**(3): 301-319.
- Schwartz, J. T. and M. Sharir (1983). "On the "piano movers" problem I. The case of a two - dimensional rigid polygonal body moving amidst polygonal barriers." Communications on pure and applied mathematics **36**(3): 345-398.
- Seth, A., J. M. Vance and J. H. Oliver (2011). "Virtual reality for assembly methods prototyping: a review." Virtual reality **15**(1): 5-20.
- Shellshear, E., S. Tafuri and J. Carlson (2014). "A multi-threaded algorithm for computing the largest non-colliding moving geometry." Computer-Aided Design **49**(0): 1-7.
- Simón, T., J.-P. Laumond, C. Van Geem and J. Cortes (2001). Computer aided motion: Move3D within MOLOG. Proceedings of IEEE International Conference on Robotics and Automation, IEEE.
- Spitz, S. N. and A. A. Requicha (2000). "Accessibility analysis using computer graphics hardware." IEEE Transactions on Visualization and Computer Graphics **6**(3): 208-219.
- Spitz, S. N., A. J. Spyridi and A. A. Requicha (1999). "Accessibility analysis for planning of dimensional inspection with coordinate measuring machines." IEEE Transactions on Robotics and Automation **15**(4): 714-727.
- Spyridi, A. and A. G. Requicha (1991). Accessibility Analysis for Polyhedral Objects. Engineering Systems with Intelligence. S. Tzafestas, Springer Netherlands. **9**: 317-324.
- Spyridi, A. J. (1994). Automatic generation of high level inspection plans for coordinate measuring machines, University of Southern California.
- Sun, Z., D. Hsu, T. Jiang, H. Kurniawati and J. H. Reif (2005). "Narrow passage sampling for probabilistic roadmap planning." IEEE Transactions on Robotics **21**(6): 1105-1115.
- Taħ, M., D. Flavigné and E. Ferré (2012). "Human interaction with motion planning algorithm." Journal of Intelligent & Robotic Systems **67**(3-4): 285-306.
- Tapia, L., S. Thomas, B. Boyd and N. M. Amato (2009). An unsupervised adaptive strategy for constructing probabilistic roadmaps. Proceedings of IEEE International Conference on Robotics and Automation, ICRA, IEEE.
- Vale, A., D. Fonte, F. Valente and I. Ribeiro (2014). "Trajectory optimization for autonomous mobile robots in ITER." Robotics and Autonomous Systems **62**(6): 871-888.

- Varadhan, G., Y. J. Kim, S. Krishnan and D. Manocha (2006). Topology preserving approximation of free configuration space. Proceedings of IEEE International Conference on Robotics and Automation ICRA'06 IEEE.
- Varadhan, G., S. Krishnan, T. Sriram and D. Manocha (2005). "A simple algorithm for complete motion planning of translating polyhedral robots." The International Journal of Robotics Research **24**(11): 983-995.
- Varadhan, G. and D. Manocha (2005). Star-shaped Roadmaps-A Deterministic Sampling Approach for Complete Motion Planning. Robotics: Science and Systems, Citeseer.
- Vázquez Hurtado, C. and J. Rosell Gratacòs (2008). Haptic guidance based on harmonic functions for the execution of teleoperated assembly tasks. Preprints of the 2007 IFAC Workshop on Intelligent Assembly and Disassembly: IAD'07, International Federation of Automatic Control.
- Vélez, Y., A. Lozano-Rodero, A. Suescun and T. Gutiérrez (2013). "Natural and hybrid bimanual interaction for virtual assembly tasks." Virtual Reality: 1-11.
- Vlachos, M., M. Hadjieleftheriou, D. Gunopulos and E. Keogh (2003). Indexing multi-dimensional time-series with support for multiple distance measures. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.
- Vlachos, M., M. Hadjieleftheriou, D. Gunopulos and E. Keogh (2006). "Indexing multidimensional time-series." The VLDB Journal—The International Journal on Very Large Data Bases **15**(1): 1-20.
- Volkov, S. and J. M. Vance (2001). "Effectiveness of haptic sensation for the evaluation of virtual prototypes." Journal of Computing and Information Science in Engineering **1**(2): 123-128.
- Walker, A. (2008). "Dubins-Curves: an open implementation of shortest paths for the forward only car." from <https://github.com/AndrewWalker/Dubins-Curves>
- Weller, R. and G. Zachmann (2009). Inner sphere trees for proximity and penetration queries. Robotics: Science and Systems.
- Weller, R. and G. Zachmann (2011). Inner Sphere Trees and Their Application to Collision Detection. Virtual Realities. G. Brunnett, S. Coquillart and G. Welch, Springer Vienna: 181-201.
- Wenger, P. and P. Chedmail (1991). "Ability of a robot to travel through its free work space in an environment with obstacles." The International journal of robotics research **10**(3): 214-227.

- Wilmarth, S. A., N. M. Amato and P. F. Stiller (1999). MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. Proceedings of IEEE International Conference on Robotics and Automation, IEEE.
- Wilmarth, S. A., N. M. Amato and P. F. Stiller (1999). Motion planning for a rigid body using random networks on the medial axis of the free space. Proceedings of the fifteenth annual symposium on Computational geometry, ACM.
- Yan, Y., E. Poirson and F. Bennis (2013). Integrating User to Minimize Assembly Path Planning Time in PLM. Product Lifecycle Management for Society, Springer Berlin Heidelberg.
- Yan, Y., E. Poirson and F. Bennis (2013). Interactive and on-line learning system for assembly task motion planning. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, Oregon, USA, ASME.
- Yan, Y., E. Poirson and F. Bennis (2015). "An interactive motion planning framework that can learn from experience." Computer-Aided Design **59**(0): 23-38.
- Yeh, H.-Y. C., J. Denny, A. Lindsey, S. Thomas and N. M. Amato (2013). Uniformly sampling the medial axis. Technical Report. Department of Computer Science, Texas A&M University, Parasol Laboratory.
- Zhang, L. (2009). Efficient Motion Planning using Generalized Penetration Depth Computation. PhD, University of North Carolina.
- Zhang, L., X. Huang, Y. J. Kim and D. Manocha (2008). "D-plan: Efficient collision-free path computation for part removal and disassembly." Computer-Aided Design and Applications **5**(6): 774-786.
- Zhang, L., Y. J. Kim and D. Manocha (2007). A Fast and Practical Algorithm for Generalized Penetration Depth Computation. Robotics: science and systems.
- Zhang, L., Y. J. Kim and D. Manocha (2007). A hybrid approach for complete motion planning. Proceedings of International Conference on Intelligent Robots and Systems, IROS'07, IEEE.
- Zhang, L., Y. J. Kim and D. Manocha (2008). "Efficient cell labelling and path non-existence computation using c-obstacle query." The International Journal of Robotics Research **27**(11-12): 1246-1257.
- Zhang, L., Y. J. Kim and D. Manocha (2008). A simple path non-existence algorithm using c-obstacle query. Algorithmic Foundation of Robotics VII, Springer: 269-284.
- Zhang, L., Y. J. Kim, G. Varadhan and D. Manocha (2007). "Generalized penetration depth computation." Computer-Aided Design **39**(8): 625-638.

Zhang, L. and D. Manocha (2008). An efficient retraction-based RRT planner. Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE.

Zhang, X., Y. J. Kim and D. Manocha (2014). "Continuous penetration depth." Computer-Aided Design **46**(0): 3-13.

Zheng, S., D. Hsu, J. Tingting, H. Kurniawati and J. H. Reif (2005). "Narrow passage sampling for probabilistic roadmap planning." IEEE Transactions on Robotics **21**(6): 1105-1115.

# Personal Publications

## Journal Publication

Y. Yan, E. Poirson, F. Bennis, An interactive motion planning framework that can learn from experience, *Computer-Aided Design*, 59 (2015) 23-38.

## Conference Publication

Y. Yan, E. Poirson, F. Bennis, Interactive and on-line learning system for assembly task motion planning, in: *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2013)*, Portland, Oregon, USA, 2013, pp. V004T005A002.

Y. Yan, E. Poirson, F. Bennis, Integrating User to Minimize Assembly Path Planning Time in PLM, in: A. Bernard, L. Rivest, D. Dutta (Eds.), *Proceedings of the IFIP WG5.1 10th International Conference on Product Lifecycle Management (PLM13)*, Product Lifecycle Management for Society, Springer Berlin Heidelberg, 2013, pp. 471-480.