Léonard Jaillet
Thierry Siméon
LAAS-CNRS,
University of Toulouse,
Toulouse, France
nic@laas.fr

Path Deformation Roadmaps: Compact Graphs with Useful Cycles for Motion Planning

Abstract

In this paper we describe a new approach to sampling-based motion planning with Probabilistic Roadmap Planner (PRM) methods. Our aim is to compute good quality roadmaps which encode the multiple connectedness of the configuration space inside small but yet representative graphs which capture the different varieties of free paths well. The proposed Path Deformation Roadmaps (PDRs) rely on a notion of path deformability indicating whether or not a given path can be continuously deformed into another existing path. By considering a simpler form of deformation than that allowed between homotopic paths, we propose a method that extends the Visibility-PRM technique to construct compact roadmaps which encode richer and more suitable information than representative paths of the homotopy classes. PDRs contain additional useful cycles between paths in the same homotopy class that can be hardly deformed into each other. Experimental results show that the technique enables small roadmaps to reliably capture the multiple connectedness of complex spaces in various problems involving free-flying and articulated robots in both two- and three-dimensional environments.

KEY WORDS—Path planning, probabilistic roadmaps, path deformation

一种扩展可见性-prm技术来构造紧凑路线图的方法，该方法比同伦类的代表性路径编码更丰富、更合适的信息。

1. Introduction

Robot motion planning has led to active research over recent decades (Latombe 1991). Sampling-based approaches have now emerged as a general and effective framework for solving challenging problems that remained out of reach of the previously existing complete algorithms. Nowadays, they make it possible to handle the complexity of many practical problems arising in various fields such as robotics, graphic animation, virtual prototyping and computational biology.

The Probabilistic Roadmap Planner (PRM) introduced by Kavraki et al. (1996) and further developed in many other works (see Choset et al. 2005 and LaValle 2004 for a survey) has been shown to perform well for a broad class of multiple-query problems, including high-dimensional configuration spaces. The overall principle of PRM is to capture the connectivity of the collision-free space ($\mathcal{C}_{\text{free}}$) by a set of one-dimensional curves stored in a precomputed roadmap. The roadmap is obtained by sampling robot configurations and subsequently connecting promising samples with valid local paths generated by a simple and fast local planner. Then, multiple path planning queries can be answered efficiently by simply connecting the query configurations and searching for a solution path in the enlarged roadmap (generally smoothed in a post-processing step to improve the quality of the solution). While PRM is successful for robots with many degrees of freedom and complete in probability, its performance degrades in the presence of narrow passages which require a prohibitively high-density roadmap. A number of variants and extensions have been proposed to alleviate this problem and improve PRM performance, for example, biasing sampling around obstacles (Amato et al. 1998; Boor et al. 1999; Hsu et al. 2003) or towards the medial axis (Wilmarth et al. 1999; Holleman and Kavraki 2000; Lien et al. 2003), using free-space dilatation (Saha and Latombe 2005; Cheng et al. 2006), visibility-based filtering (Simeon et al. 2000) or adaptive sampling (Kurniawati and Hsu 2006; Rodriguez et al. 2006), exploiting search-space information (Burns and Brock 2005) or delaying collision checks (Bohlin and Kavraki 2000; Sanchez and Latombe 2002).

The International Journal of Robotics Research
Vol. 27, No. 11–12, November/December 2008, pp. 1175–1188
DOI: 10.1177/0278364908098411

Figures 1, 3–8, 12–15, 17–21 appear in color online: http://ijr.sagepub.com

While most PRM variants focus on the fast computation of roadmaps reflecting the connectivity of the free configuration space, only a few works (Schmitzberger et al. 2002; Nieuwenhuisen and Overmars 2004; Geraerts and Overmars 2006) address the problem of computing good quality roadmaps which encode the multiple connectedness of the space inside small graphs and with a limited number of useful cycles (i.e. cycles representative of the varieties[1] of free paths). PRM often leads to dense roadmaps, whereas small graphs may be considered in order to speed up both query time and roadmap updates. In contrast, Visibility-PRM (Simeon et al. 2000) produces very small roadmaps by rejecting most samples that lie in the visibility regions of existing guards. However, this pruning strategy leads to tree-like roadmaps which do not capture the multiple connectedness of the space. Introducing cycles is important to obtain higher quality solutions when postprocessing queries. It avoids the computation of unnecessarily long paths, difficult to shorten by the smoothing techniques (see, for example, Sekhavat et al. 1998 and Sanchez and Latombe 2002). Useful cycles also make the roadmap more robust to dynamic changes in the environment and may allow the planner to choose alternative routes to avoid repetitive motions (Nieuwenhuisen and Overmars 2004).

Intuitively, the probability of a roadmap capturing the different path varieties of the free configuration space well increases with its degree of redundancy. However, a direct approach attempting connections between every pair of nodes is far too costly. Thus, several heuristic-based connection strategies are usually applied to limit the number of redundant cycles. One strategy (see, for example, Kavraki et al. 1996) would be to restrict the connection attempts of new samples to the k nearest nodes of the roadmap (or of each connected component). Another approach would be to only consider nodes within a ball of radius r centered around the new sampled configuration (see, for example, Bohlin and Kavraki 2000). A more recent technique proposed by Nieuwenhuisen and Overmars (2004) creates cycles only between already connected nodes if they are k times more distant in the roadmap than in the configuration space. This idea is also used by Geraerts and Overmars (2006) to create high-quality roadmaps for simple two or three degree of freedom (DOF) robots in virtual environments. In all cases, the capture of relevant path varieties notably varies depending on the choice of some parameters (e.g. k or r). Using these heuristic sampling strategies, accurate parameter values for a given environment are difficult to define. It may result in a significant loss of performance of the roadmap construction process. A more formal technique (Schmitzberger et al. 2002) proposed for two-dimensional problems only considers cycles that encode homotopy classes of the free space. Other related works aim to increase the roadmap connectivity in constrained directions of the configuration space using a node connection strategy based on a Delaunay triangulation (Huang and Gupta 2004). Finally, some authors exploit cycles to provide alternative routes in dynamic environments with mobile obstacles (van den Berg et al. 2005).

In the present article, we develop a new method to build compact roadmaps which are yet representative of the different varieties of free paths. The method only generates a limited number of useful cycles in the roadmap. Moreover, the algorithm stops automatically when most of the relevant alternative paths have been found. Our approach relies on the notion of "path deformability", indicating whether or not a given path can be deformed continuously into another existing path. Compared with the standard notion of homotopy (which is not directly suitable for our purpose because it relies on excessively complex deformations; see Section 2), we consider simpler and more easily computable deformations between paths (Section 3). This results in compact roadmaps that capture a richer set of paths than homotopy (Section 4). We describe in Section 5 a two-stage algorithm to construct such (easy) Path Deformation Roadmaps (PDRs). The first stage is based on Visibility-PRM (Simeon et al. 2000) to construct a small tree covering the space and capturing its connected components as well as possible. The second stage aims at enriching the roadmap with new nodes involved in the creation of useful cycles. The key ingredient of this step is an efficient path visibility test used to filter useless paths which can be easily deformed into existing roadmap paths. Following the philosophy of Visibility-PRM, the second stage also integrates a stop condition based on the difficulty of finding new useful cycles. Finally, some experimental results (Section 6) show that the technique enables small roadmaps to reliably capture the multiple-connectedness of configuration spaces in various problems involving free-flying or articulated robots.

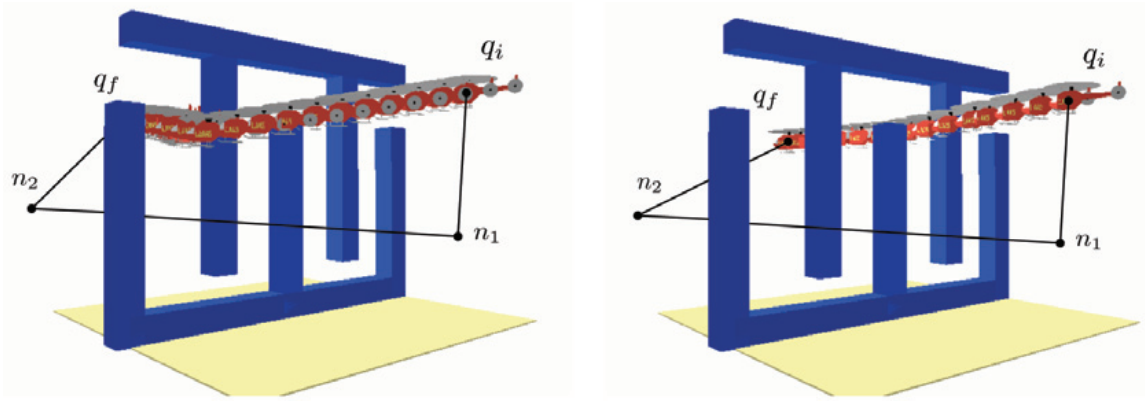
第一阶段是基于可见度-PRM (Simeon等人，2000年)，以构建一个小树覆盖空间，并尽可能捕捉其连接的组件。第二个阶段旨在用创建有用周期所涉及的新节点来丰富路线图。这一步的关键要素是有效的路径可见性测试，用于过滤无用的路径。这些路径很容易变形为现有的路线图路径。遵循PRM可见度的哲学，第二阶段还整合了一个基于难以找到新的有用周期的停止条件。

2. Homotopy Versus Useful Roadmap Paths

First of all, we discuss the relation between homotopy and representative path varieties that it would be desirable to store in the roadmap. The capture of the homotopy classes of $\mathcal{C}_{\text{free}}$ corresponds to a stronger property than connectivity. Two paths are called homotopic (with fixed endpoints) if one can be "continuously deformed" into the other (see Section 3.1). Homotopy defines an equivalence relation on the set of all paths of $\mathcal{C}_{\text{free}}$. A roadmap capturing the homotopy classes means that every valid path (even cyclic paths) can be deformed continuously into a path of the roadmap. PRM methods usually do not ensure this property. Only the work of Schmitzberger et al. (2002) considers the problem formally and sketches a method to encode the set of homotopy classes inside a probabilistic roadmap. However, the approach is only applied to two-dimensional problems and its extension is limited by the

捕获同伦类的路线图意味着每一条有效的路径（甚至是循环路径）都可以连续地变形为路线图的一条路径。PRM方法通常不能确保此属性。只有施密茨伯格等人的作品。

1. The term "path variety" is used in this paper to refer to a given class of similar paths.



(a) (b)

Fig. 1. Two examples of query for a two-node graph (n_1–n_2). In (a) the solution path (q_i–n_1–n_2–q_f) extracted from the graph could be easily deformed into the displayed short path connecting query configurations (q_i, q_g) whereas a deformation in $\mathcal{C}_{\text{free}}$ would be much more complex in (b).

difficulty of characterizing homotopic deformations in higher dimensions.

Moreover, as was revealed by Nieuwenhuisen and Overmars (2004), capturing the homotopy classes in higher dimensions may not be sufficient to encode the set of representative paths since homotopic paths (i.e. paths in the same homotopy class) may be too hard to deform into each other. This problem is illustrated by the example in Figure 1. Here, $\mathcal{C}_{\text{free}}$ contains only one homotopy class. Therefore, a homotopy-based roadmap would have a tree structure, such as the simple two-node (n_1–n_2) tree shown in Figure 1. While for the query example in Figure 1(a), the solution path (q_i–n_1–n_2–q_f) found in the roadmap could be easily deformed into the displayed short path connecting query configurations (q_i, q_g), a free deformation would be much more difficult to compute for the example in Figure 1(b). Even if the topological nature of the two displayed paths is the same, their difference leads to store a representation of both paths in the roadmap. More generally, one can say that a roadmap is a good representation of free path varieties if any path can be "easily" deformed into a path of the roadmap. This notion of simple path deformation is formalized below.

即使这两个显示的路径的拓扑性质是相同的，它们的差异也会导致在路线图中存储这两条路径的表示。更一般地说，人们可以说，如果任何路径都可以“容易”变形为路线图的路径，那么路线图是自由路径品种的良好表示。简单路径变形的概念如下形式化。

3. Complexity of a Path Deformation

In this section, after briefly recalling the definition of a homotopic deformation, we propose a way to characterize classes of path deformations according to their complexity.

3.1. Homotopy

The homotopy between two paths is a standard notion from topology (see Hatcher (2002) for a complete definition). Two paths τ and τ' in a topological space X are *homotopic* (with fixed endpoints) if there exists a continuous map $h : [0,1] \times [0,1] \to X$ with $h(s,0) = \tau(s)$ and $h(s,1) = \tau'(s)$ for all $s \in [0,1]$ and $h(0,t) = h(0,0)$ and $h(1,t) = h(1,0)$ for all $t \in [0,1]$.

Homotopy is a way to define any continuous deformation from one path to another. In the following, we introduce a less general class of deformations, called *K-order deformations*, characterizing particular subsets of homotopic deformations. We use K-order deformations in Section 4 to compute PDRs.

3.1. K-order Deformation

Definition 1. A *K-order deformation* is a particular homotopic deformation such that each curve transforming a point of τ into a point of τ' is an angle line of K segments, that is, a piecewise linear curve formed by K consecutive straight-line segments. 是一种由K个连续直线段形成的分段线性曲线

Therefore, a first-order deformation surface describes a ruled surface[2] and a K-order deformation is obtained by concatenation of K ruled surfaces. This is illustrated in Figure 2, which depicts different types of path deformations: Figure 2(a) is a general homotopic deformation whereas Figure 2(b) and (c) are first-order and second-order deformations respectively.

通过K规则曲面的连接得到了k阶变形。

Let D_i denote the set of i-order deformations. We clearly have $D^i \subset D^j$ for all $i < j$. Thus, the K value of the smallest K-order deformation existing between two paths is a good measure of the difficulty to deform one path into the other.

在两条路径之间存在的最小k阶变形的K值可以很好地衡量了将一条路径变形成另一条路径的困难。

2. A ruled surface is a surface that can be swept out by moving a straight line in space.

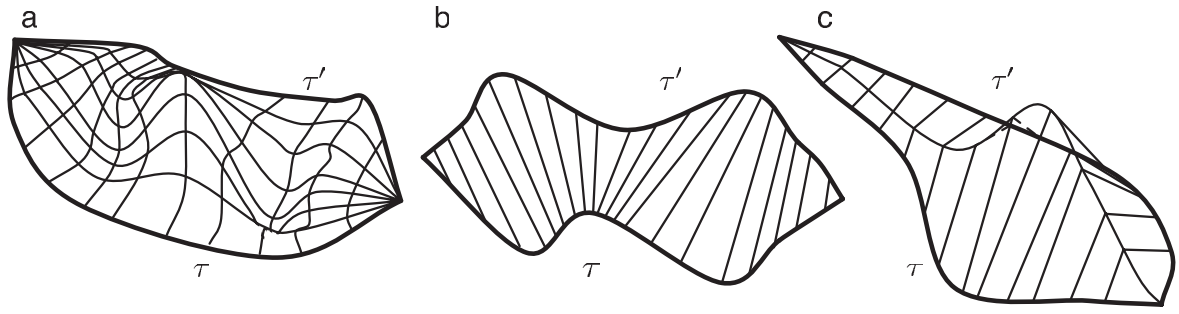


Fig. 2. (a) General homotopic deformation. (b) First-order deformation: the deformation surface is a ruled surface. (c) Second-order deformation: the deformation surface is obtained by concatenating two ruled surfaces.

变形曲面是通过连接两个规则曲面来得到的。

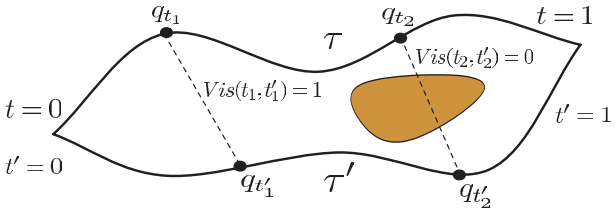


Fig. 3. The parametric visibility function of two paths evaluates the visibility between the points of each path.

3.2. Visibility Diagram of Paths

当且仅当每个路径在每点保持可见性约束的同时，两条路径之间存在一阶变形

It is important to note that a first-order deformation between two paths exists if and only if it is possible to simultaneously go through the two paths while maintaining a visibility constraint between the points of each path (see Figure 3). This formulation provides a computational way to test the existence of a first-order deformation, also called *visibility deformation* between two paths. Let $\mathcal{L}_{\text{lin}}$ be the straight-line segment between two configurations of $\mathcal{C}$. The parametric visibility function *Vis* of two paths (τ, τ') is defined as follows:

$$Vis: \begin{cases} [0,1]\times[0,1] & \rightarrow & \{0,1\}, \\ Vis(t,t') & = & 1 \text{ if } \mathcal{L}_{\text{lin}}(\tau(t),\tau'(t')) \in \mathcal{C}_{\text{free}}, \\ Vis(t,t') & = & 0 \text{ otherwise.} \end{cases}$$

Then, the visibility diagram of paths (τ, τ') is defined as the two-dimensional diagram of the *Vis* function. This is illustrated in Figure 4, which depicts several examples of computed visibility diagrams with the corresponding paths. The visibility (i.e. first-order) deformation between two paths can now be expressed as follows: two paths (τ, τ') (with the same endpoints) are *visibility deformable* one into the other if and only if there is a path in their visibility diagram linking points of parameters $(0, 0)$ and $(1, 1)$. Therefore, it is possible to test the visibility deformation between two paths by computing their visibility diagram and searching for a path in the diagram linking points $(0, 0)$ and $(1, 1)$.

In the first two examples, Figure 4(a) and (b), there is no visibility deformation between the paths (τ, τ') since obstacles inside the cycle paths forbid any homotopic deformation. In the third example, Figure 4(c), a homotopic deformation between τ and τ' is possible, but the two paths are still not deformable by visibility. Finally, a visibility deformation is possible for the last example, Figure 4(d), where a valid path linking points $(0, 0)$ and $(1, 1)$ can be found in the visibility diagram.

4. K-order Deformation Roadmap

在前一节中，我们定义了一种方法来描述两个路径一个变形为另一条的复杂性。这种形式现在被用于定义给路线图捕获配置空间中不同种类的自由路径的能力。

In the previous section we have defined a way to characterize the complexity for two paths to be deformed one into the other. This formalism is now used to define the ability of a given roadmap to capture different varieties of free paths of the configuration space.

Definition 2. A roadmap R is a *K-order deformation roadmap* if and only if for any path τ of $\mathcal{C}_{\text{free}}$ it is possible to extract a path τ'_R from R (by connecting the two extreme configurations of the path) such that τ and τ'_R are K-deformable.

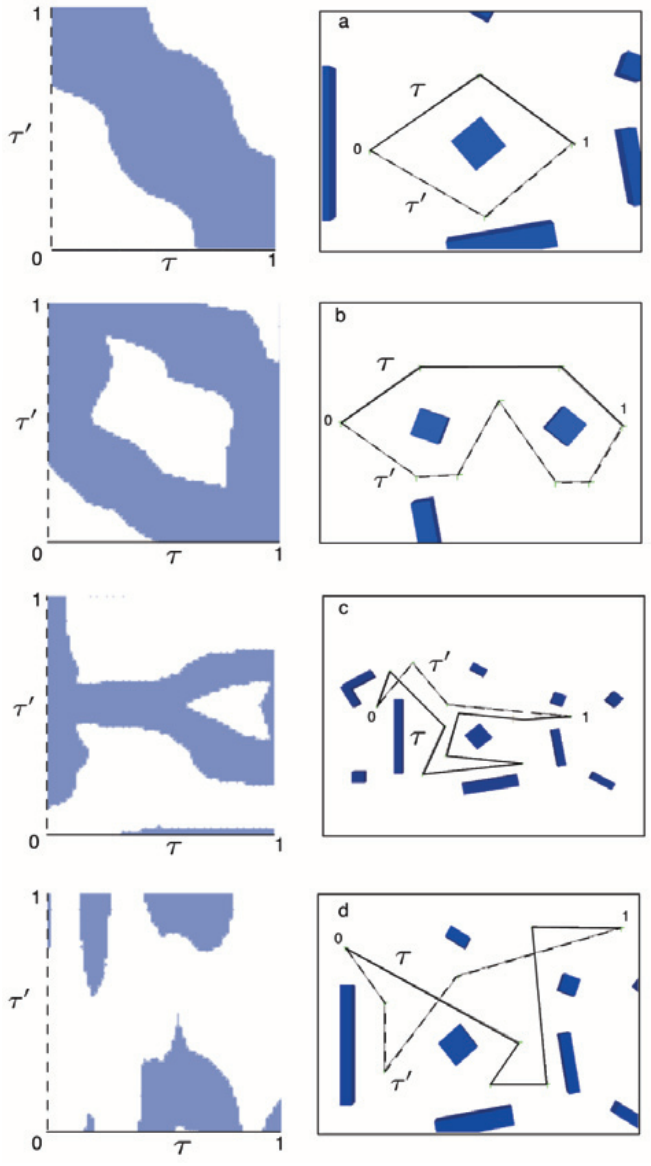
This definition establishes a strong criterion specifying how different varieties of free paths are captured inside the roadmap. Since a K-order deformation is a specific kind of homotopic transformation, any deformation roadmap captures the homotopy classes of $\mathcal{C}_{\text{free}}$. The following sections present a computational method to construct such roadmaps.

4.1. Visibility Deformation Roadmap

In the following, we first define the notion of a *Roadmap Connected from any Point of View* (called a *RCPV* roadmaps) pre-

从任何角度连接的路线图

因为循环路径内的障碍物禁止出现任何同伦变形



其中在可见性图中可以找到有效的路径链接点

Fig. 4. Visibility diagrams for pairs of paths τ, τ' with the same endpoints. White areas represent regions where $Vis(t, t') = 1$. A visibility deformation is only possible in (d), where a valid path linking points $(0, 0)$ and $(1, 1)$ can be found in the visibility diagram.

viously introduced by Schmitzberger et al. (2002). Then, we establish that RCPV roadmaps are visibility (i.e. first-order) deformation roadmaps.

4.1.1. Visible Subroadmap

Let R be a roadmap with a set of nodes N and a set of edges E. Let also assume that R covers $\mathcal{C}_{\text{free}}$. The coverage property means that every configuration in $\mathcal{C}_{\text{free}}$ is visible from a node of R. Thus, it is possible to extract from N a subset of nodes G (called guards) sufficient to maintain this coverage. Then, for a free configuration q_v, we can define the *Visible Subroadmap* $R_v = (N_v, E_v)$, as follows:

- N_v, a sublist of guards visible from q_v, $N_v = \{g \in G / \mathcal{L}_{\text{lin}}(q_v, g) \in \mathcal{C}_{\text{free}}\}$;
- E_v, a sublist of edges visible from q_v, $E_v = \{e \in E / \mathcal{L}_{\text{lin}}(q_v, e) \in \mathcal{C}_{\text{free}}\}$.

Note that the notation $\mathcal{L}_{\text{lin}}(q_v, e) \in \mathcal{C}_{\text{free}}$ means that $\{\forall q \in e, \mathcal{L}(q_v, q) \in \mathcal{C}_{free}\}$. Examples of visible subroadmaps are presented in Figure 5.

4.1.2. RCPV Roadmaps

Definition 3. A RCPV roadmap is such that for any configuration of $\mathcal{C}_{\text{free}}$, the visible subroadmap is connected.

The following property establishes the link between RCPV roadmaps and visibility deformation roadmaps.

Property 1. *A RCPV roadmap is a particular case of visibility deformation roadmap.* RCPV路线图是可见性变形路线图的一个特殊情况

Proof. Sketch of proof Let R be a RCPV roadmap, and τ a path of $\mathcal{C}_{\text{free}}$. As a RCPV roadmap ensures the coverage of $\mathcal{C}_{\text{free}}$, τ can be covered by a given set of n guards, inducing its partitioning into successive elementary paths:

$$\tau = \{\tau_{g_1} \oplus \tau_{g_1 \cap g_2} \oplus \cdots \oplus \tau_{g_i} \oplus \tau_{g_i \cap g_{i+1}} \oplus \tau_{g_{i+1}} \oplus \cdots \oplus \tau_{g_{n-1}} \oplus \tau_{g_{n-1} \cap g_n} \oplus \tau_{g_n}\},$$

with τ_{g_i} denoting the portion of path visible from the g_i guard and $\tau_{g_i \cap g_{i+1}}$ the portion visible simultaneously from g_i and g_{i+1} (cf. Figure 6). Note that τ can possibly go through the visibility region of a guard multiple times. Thus, we can have $g_i = g_j$ for $j \neq i - 1$ and $j \neq i + 1$ and τ is therefore partitioned into at least $2n - 1$ portions.

Since τ_{g_i} and g_i are visible by definition, it is possible to build a patch of ruled surface between them (Figure 7(a)). Similarly, there is a patch of ruled surface between $\tau_{g_{i+1}}$ and g_{i+1}. As R is a RCPV roadmap, any configuration $q_v \in \tau_{g_i} \cap \tau_{g_{i+1}}$ sees a path τ'_R connecting g_i to g_{i+1}. This property makes it possible to build a third patch of ruled surface between q_v and τ'_R (Figure 7(b)). Finally, it is possible to fuse these three patches into a single ruled surface between $\tau_{g_i} \cap \tau_{g_{i+1}}$ and τ'_R (Figure 7(c)). Thus, there exists a ruled surface (i.e. a visibility deformation surface) between the totality of τ and a path of the roadmap. □

RCPV roadmaps are first-order deformation roadmaps. However, these roadmaps involve a high level of redundancy

RCPV路线图是一阶变形路线图。然而，这些路线图涉及到高水平的冗余

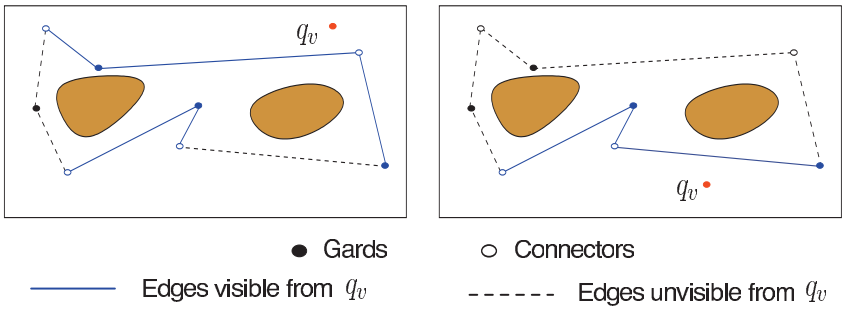


Fig. 5. Two examples of visible subroadmaps from a given configuration q_v. In (a), the visible subroadmap is disconnected, whereas it is connected in (b).

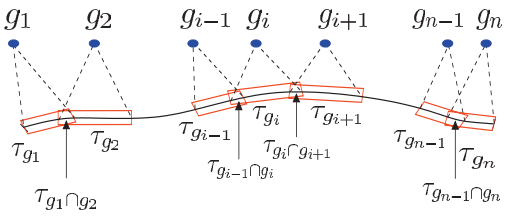


Fig. 6. Path decomposition as a function of the portions visible from the guard nodes.

路径分解作为从保护节点中可见的部分的函数。

(see Section 6) and yet contain many useless cycles, especially in constrained situations. Therefore, to keep a compact structure, we filter a part of the redundancy as explained in the following section. We show that this filtering leads to a second-order deformation roadmap.

这种过滤导致了一个二阶变形路线图。

4.2. Second-order Deformation Roadmaps

Let $R = (N, E)$ be a RCPV roadmap and $G \in N$ be a set of guard nodes ensuring the $\mathcal{C}_{\text{free}}$ coverage. Consider a given pair of guards and τ, τ' two paths of the roadmap (1) linking these guards (i.e. creating a cycle) and (2) visibility deformable one into the other. Then, we have the following property.

能见度可变形为路径，并连接相同的防护装置

Property 2. *From a RCPV roadmap R, deletion of redundant paths τ'_R (i.e. visibility deformable into path τ and connecting the same guards) leads to a second order deformation roadmap.*

Proof. Sketch of proof Consider the partition of a free path τ, as defined in Section 4.1.2. In that section we have shown that with a RCPV roadmap, one can extract a roadmap path τ'_R such that $\tau_{g_i} \cap \tau_{g_{i+1}}$ is visibility deformable into τ'_R (Figure 8(a)). Now, let suppose that the redundant path τ'_R has been deleted as proposed above. It means that τ'_R was visibility deformable into another path τ''_R which remains in the roadmap

通过连接两个规则曲面，可以在任何 自由路径和路线图路径之间建立一个二阶变形曲面

(Figure 8b). Thus, by concatenation of two ruled surfaces it is possible to build a second-order deformation surface between any path τ of $\mathcal{C}_{\text{free}}$ and a path of the roadmap (Figure 8(c)). □

The above proofs are not constructive. The next section describes a sampling-based algorithm to construct non-redundant graphs (referred to as "Path Deformation Roadmaps" in the rest of this article) which tend to satisfy the second-order path deformation property.

Path Deformation Roadmaps

5. Algorithm to Build PDRs

The algorithm proposed to construct PDRs proceeds in two stages. First, it computes a small covering tree that captures the connectedness of the space. During a second stage, the initial tree is enhanced with useful cycles required for the multiple connectedness.

首先，它计算一个捕获空间连接性的小覆盖树。在第二阶段，初始树被增强为多个连接所需的有用周期。

The initial covering tree is computed using Visibility-PRM. The pseudocode of the algorithm is shown in Figure 9 (see Simeon et al. (2000) for a detailed description). Visibility roadmaps rely on a free-space structuring into visibility domains (i.e. sets of configurations connectable to a given guard by a valid local path). Computed guards are linked together via *connectors* located in their overlapping visibility regions. Such roadmaps can be constructed using a simple PRM variant: each free sample is added to the roadmap only if it cannot be connected to any existing node (i.e. guard) or if it connects at least two components (i.e. connector). The end of the algorithm is controlled by the difficulty of adding a new guard ($ntry_{\max}$ parameter) which relates to the quality of the roadmap in term of coverage (Simeon et al. 2000). The computed roadmap is a small tree (i.e. no cycles) capturing the free space coverage with a limited number of nodes and edges. However, there is no guarantee at this stage concerning the deformability of $\mathcal{C}_{\text{free}}$ paths into roadmap paths.

Therefore, the visibility tree is enriched with nodes and edges during the second stage, creating the useful cycles required to obtain a PDR. Instead of building a RCPV roadmap

因此，在第二阶段使用节点和边丰富可见性树，从而创建获得PDR所需的有用周期。而不是构建一个RCPV路线图

可见性路线图依赖于一个进入可见性领域的自由空间结构。可通过有效的本地路径连接到给定防护装置的配置集）。计算机防护装置通过位于其重叠可见性区域的连接器连接在一起。这样的路线图可以使用一个简单的PRM变体来构建：每个免费的样本只有在不能连接到任何现有的节点时才会被添加到路线图中。或者它至少连接了两个组件。（连接接头）。算法的结束是由难以添加一个新的保护器(ntrymax参数)来控制的，它与路线图的覆盖质量有关(Simeon等人)。2000)。计算出的路线图是一个小树。没有循环)用有限数量的节点和边缘捕获自由空间覆盖。然而，在这个阶段，并不能保证 自由路径进入路线图路径的变形性。

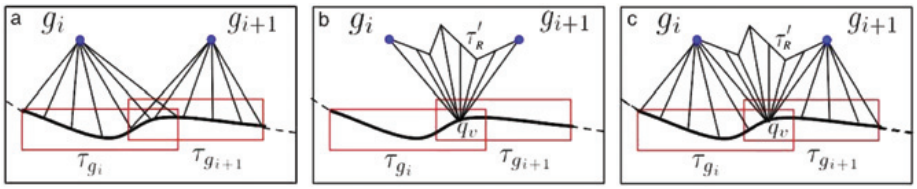


Fig. 7. A RCPV roadmap is a visibility deformation roadmap. (a) The visibility of the guard gives the first patches of ruled surfaces. (b) The RCPV roadmap property guarantees the visibility of a roadmap path connecting two guards. (c) By construction, a global visibility deformation surface can be built.

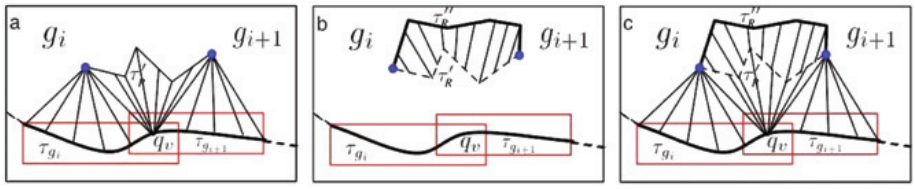


Fig. 8. Deleting redundant paths in a RCPV roadmap leads to a second-order deformation roadmap. (a) Visibility deformation between a path τ and a RCPV roadmap path τ'_R. (b) A filtered path τ'_R is visibility deformable into a roadmap path τ''_R. (c) By construction, there is a second-order deformation surface between a free path and a portion of roadmap.

算法在路线图中添加一个新循环之前直接执行冗余滤波

first, and then filtering the redundant cycles (as defined in Section 4.2), the algorithm performs the redundancy filtering directly before each addition of a new cycle to the roadmap (for efficiency purpose).

The pseudocode of the algorithm used to build a PDR is shown in Figure 10. At each iteration, a free configuration q_v is randomly sampled and the connectivity of the visible subroadmap R_v seen by the sample is determined (*TestVisibSubRoadmap* function line 6). When the visible subroadmap is found to be singly connected, the sample can be rejected directly since any connection to the roadmap will obviously yield useless cycles (see Figure 5(b)). In any other case, a redundancy test of cycle paths possibly created by q_v must be performed. As explained in Section 5.1, the connectivity test of the visible subroadmap stops as soon as the subroadmap is found to be disconnected (avoiding, as much as possible, a whole connectivity test) and returns the two computed components $Comp_1$ and $Comp_2$. Then, the nearest nodes n_1, n_2 from q_v are selected among these components and are used to test whether there is a visibility deformation between the path $\tau = n_1 - q_v - n_2$ and a roadmap path linking n_1 to n_2 (*TestRedundancy* function line 10). If such a visibility deformation exists, the configuration is useless with regards to the construction of a second-order deformation roadmap and is therefore rejected. Otherwise, q_v is inserted in the roadmap as a new node and $n_1 - q_v, n_2 - q_v$ are also inserted as new edges. The algorithm memorizes the number of successive failures since the last useful cycle inserted. Similarly to the termination control of Visibility-PRM, this information is used to stop the iterations when the insertion of a new cycle becomes too difficult, that is, when most of the useful cycles are already captured. Dealing with convergence, a roadmap computed with Path-Deformation-PRM tends toward a second-order deformation roadmap for sufficiently high values of its termination control parameter ($ntry\_cycl_{\max}$).

当发现可见子路线图是单连接时，可以直接拒绝样本，因为任何与路线图的连接都会产生无用的循环

In the following, we provide details of the algorithms used to establish the subroadmap connectivity (*TestVisibSubRoadmap* function) and to test the visibility deformation between pairs of paths (*TestRedundancy* function).

5.1. Visible Subroadmap

The pseudocode of the *TestVisibSubRoadmap* function (Figure 11) outlines the lazy evaluation method used to check the connectivity of a visible subroadmap seen from a given configuration q_v. This two-stage process is illustrated in Figure 12. Starting from the current roadmap (Figure 12(a)), every edge is first initialized as potentially visible. The algorithm first checks the node visibility from q_v by testing the collision-freeness of straight-line segments linking q_v to each roadmap node (Figure 12(b)). The set of non-visible nodes is then used to speed up the connectivity test. Indeed, when a given node is labeled as non-visible, all of its edges can also be labeled as non-visible from q_v. In most cases, this fast test is sufficient to establish the disconnectedness of the visible subroadmap without requiring more costly tests. Otherwise, the algorithm further proceeds by computing the visibility of edges linking the

VISIBILITY-PRM

input: the robot A, the environment B, $ntry_{max}$

output: a roadmap R with a tree structure

```
1   ntry ← 0
2   While ntry < ntry_max
3       q ← RandomFreeConfig(A, B)
4       g_vis ← ∅; Connector ← False
5       For all components R_i of R
6           g ← VisibleConfInComponent(q, R_i)
7           If g ≠ ∅
8               if g_vis = ∅
9                   g_vis ← g
10              Else
11                  NewConnector(q, g, g_vis)
12                  Connector ← True
13              End If
14          End If
15      until Connector = True
16      If (g_vis = ∅)
17          NewGuard(q)
18          ntry ← 0
19      Else
20          ntry ← ntry + 1
21      End if
22  End While
```

Fig. 9. Visibility-PRM algorithm used to compute an initial tree in the PDR method.

visible nodes (Figure 12(c)). Note that edges are not systematically tested since the computation stops as soon as the visible subroadmap is found to be disconnected (Figure 12(d)). In the next section, we describe the visibility test between a whole edge and a given configuration.

5.1.1. Edge Visibility

Testing the visibility of an edge from a configuration q_v is equivalent to checking the validity of triangular configuration-space facets, defined by q_v and the two edge's endpoints (cf. Figure 13). The test can involve one or several facets depending on the topological nature of $\mathcal{C}$.

- If $\mathcal{C}$ is isomorphic to $[0, 1]^n$ (the robot's DOFs are only translations and/or bounded rotations) then the visibility test can be performed by testing only a single facet in $\mathcal{C}$ (Figure 14(a)).

PATH-DEFORMATION-PRM

input: the robot A, the environment B, $ntry_{max}$, $ntry\_cycl_{max}$

output: a PDR

```
1   R ← Visibility-PRM(A, B, ntry_max)
2   ntry_cycl ← 0
3   While ntry_cycl < ntry_cycl_max
4       q_v ← RandomFreeConfig(A, B)
5       ntry_cycl ← ntry_cycl + 1
6       If TestVisibSubRoadmap(R, q_v) = Disconnected
7           n_1 ← NearestGuard(q_v, Comp_1(R_v))
8           n_2 ← NearestGuard(q_v, Comp_2(R_v))
9           τ ← BuildPath(n_1, q_v, n_2)
10          If TestRedundancy (τ, n_1, n_2, R) = False
11              CreateCyclicPath(τ, R)
12              ntry_cycl ← 0
13          End If
14      End If
15  End While
```

Fig. 10. *General algorithm to build a PDR.*

TestVisibSubRoadmap(R, q_v)

```
1   N_vis ← EmptyList
2   For all node n ∈ R
3       If VisibleNode(n, q_v)
4           AddToList(n, N_vis)
5       End If
6   Endfor
7   TestEdges ← False
8   If VisibleConnectivity(q_v, N_vis, R, TestEdges) = False
9       Return Disconnected
10  End If
11  TestEdges ← True
12  If VisibleConnectivity(q_v, N_vis, R, TestEdges) = False
13      Return Disconnected
14  End If
15  Return Connected
```

Fig. 11. Algorithm testing the visible subroadmap connectivity from a given configuration q_v.

同形的，同构的

- If $\mathcal{C}$ is isomorphic to $[0, 1]^n \times SO(d)^m$ with $m > 0$ (i.e. one or more DOFs are cyclic), the visibility test of an edge can lead to several facets being tested (Figure 14(b)). A discontinuity leading to a split in two

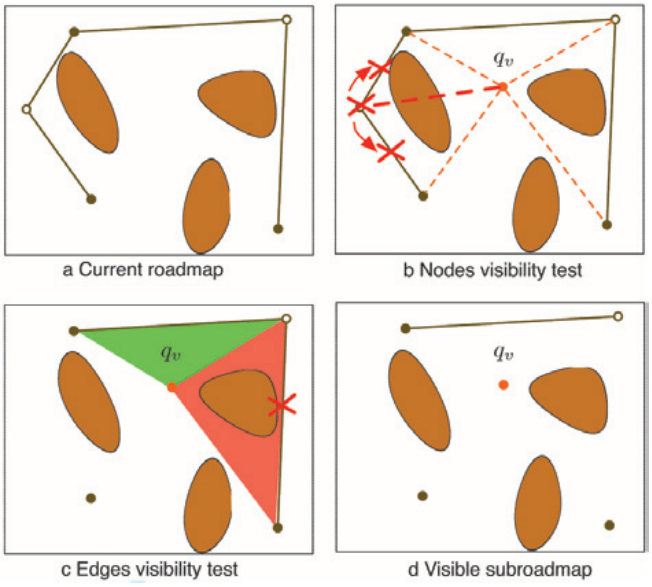


Fig. 12. Two-stage connectivity test of a visible subroadmap.

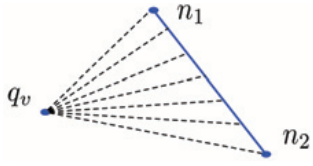


Fig. 13. Edge visibility: n_1–n_2 is visible from q_v if the facet $\{q_v, n_1, n_2\}$ is valid.

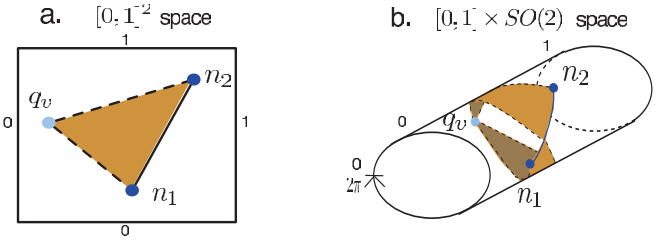


Fig. 14. Testing the visibility of an edge can lead to test one (a) or several (b) facets, depending on the topological nature of the configuration space.

facets occurs each time the distance between q_v and a configuration on the edge is equal to π, according to a given DOF.

5.1.2. Elementary Facet Test

To test the validity of a facet, we try to cover it entirely with free balls of $\mathcal{C}$ (Figure 15). First, the radii of the free balls

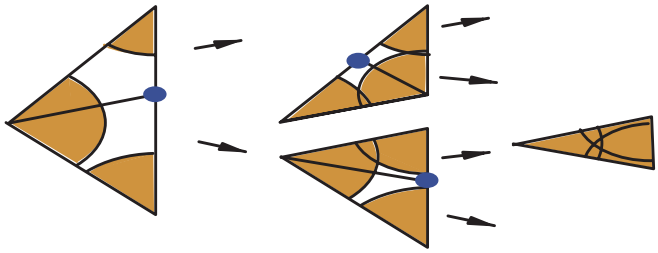


Fig. 15. Dichotomic covering of a valid facet with $\mathcal{C}_{\text{free}}$ balls.

centered on each vertex of the facet are computed. If they are sufficient to cover the facet, the algorithm returns that the facet is valid. Otherwise, it is split into two subfacets computed such that their common vertex is as far as possible from the regions already covered by the balls. The radius of the ball centered on this new vertex is then computed. This dichotomic process is performed until the entire facet is covered or until one vertex is tested as invalid.

To compute the radius of a free ball centered on a vertex, we use a conservative method based on the robot kinematics and minimal distances of its bodies to the obstacles. The principle is similar to that used for path collision detection with non-uniform step size (see LaValle (2004) for a formal presentation and Jaillet (2005) for the extension to the case of free balls of $\mathcal{C}$). In practice, such methods can be too conservative when applied to complex robots with many rotational DOFs. A discrete variant of the edge visibility test can be preferable to efficiently deal with such cases. It consists of discretizing the edge and checking the validity of the straight-line paths that connect q_v to the intermediate configurations along this edge. Another advantage of this discrete variant is to avoid the elementary facet decomposition phase (the switch of direction along the edge is performed automatically when the algorithm checks the validity of straight-line paths). Note that the discrete test was used in our experiments for the 6-DOF manipulator example (see Figure 21).

5.2. Redundancy Test

A disconnected subroadmap from the point of view of a configuration q_v can be reconnected by a path $\tau = n_1$–q_v–n_2 with n_1, n_2 belonging to two distinct subcomponents. Such connection has to be performed only if it introduces cycles that are useful with regards to the construction of a second-order deformation roadmap. Testing the usefulness of adding a path τ is performed by the *TestRedundancy* algorithm (see the pseudocode in Figure 16). Roadmap paths linking nodes n_1 and n_2 are iteratively extracted and tested according to their visibility deformation relatively to τ. This process starts with the shortest path and stops when a visibility deformation is

```
TestRedundancy(τ, n1, n2, R)
1  τ'R ← BestPath(n1, n2, R)
2  While τ'R ≠ ∅
3      If VisibDeformation(τ, τ'R) = True
4          Return True
5      End If
6      τ'R ← BestPath(n1, n2, R)
7  End While
8  Return False
```

Fig. 16. Visibility deformation test between a path τ and roadmap paths.

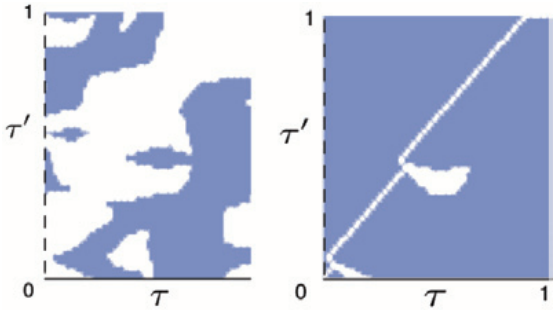


Fig. 17. (a) Visibility diagram and (b) cells explored during the visibility deformation test.

found (i.e. τ is useless and thus rejected) or when every candidate path has been tested (i.e. τ creates a useful cycle and is added to the roadmap). In practice, only the first k shortest paths found in the roadmap (e.g. $k = 10$) are considered as candidates for the redundancy test. In fact, the longest paths have less chance to be visibility deformable into the path τ. This is particularly useful for complex environments where the roadmap may contain many cycles, resulting into many paths between nodes n_1 and n_2. Finally, note that in the worst case, if a redundancy test fails to detect an existing deformation, a "useless" cycle is added but the property of second-order deformation roadmap stated in Definition 2 still holds.

The *VisibDeformation* function (line 3 of the algorithm in Figure 16) tests whether two paths τ and τ'_R can be visibility deformed one into the other. This function relies on the grid-based computation of the visibility diagram associated with the two paths. The deformation is only possible when a path between the $(0, 0)$ and $(1, 1)$ points exists in the diagram (cf. Section 3.3). In practice, the whole diagram is not computed. Tests are limited to the grid cells visited during the A^* search of a valid path in the visibility diagram, incrementally developed during the search. This implicit search of the diagram noticeably limits the number of visibility tests to be performed (Figure 17) and significantly accelerates the redundancy test. Note that further speed up may be achieved using the lazy search technique proposed by van den Berg and Overmars (2007), combined with lifelong planning A* (Koenig and Furcy 2004), aiming at further minimizing the number of grid cells tested for visibility.

6. Experimental Results

We implemented the algorithm for constructing (second-order) deformation roadmaps in the Move3D software platform (Simeon et al. 2001). The experiments reported below were performed on a 1.2 GHz G4 PowerPC running on Mac OS-X. The performance results summarized in Tables 1 and 2 correspond to average values computed over several runs of the algorithm.

The first experiment shown in Figure 18 compares the level of redundancy obtained depending on the algorithm used: (a) a minimum tree structure obtained with the Visibility-PRM, (b) a first-order roadmap (built without the filtering process) and (c) a second-order deformation roadmap that captures the different varieties of paths while maintaining a compact structure. This figure clearly reveals the interest of second-order deformation roadmaps (PDRs) over first-order (RCPV) roadmaps.

The next set of experiments (Figure 19) presents the PDRs obtained for a 2-DOF robot evolving in complex environments. The first scene (a) requires 25 elementary cycles to capture the homotopy. Our method makes it possible to build a roadmap capturing these cycles in only 109 seconds. The second scene (b) has a higher geometrical complexity (70,000 facets). The computing time (164 seconds) reported in Table 1 shows that the algorithm can efficiently handle such geometrically complex scenes. One can also note that the resulting two-dimensional roadmaps contain a very limited number of additional nodes compared with homotopy.

The third experiment (Figure 20) involves a narrow passage problem for a squared robot with three DOFs (two translations and one rotation). The robot has four ways to go through the narrow passage, depending on its orientation. Therefore, the narrow passage corresponds to four homotopy classes in the configuration space.

Table 3 depicts results obtained with a traditional k-nearest PRM (Kavraki et al. 1996) for different couples (N, k) (with N, the number of roadmap nodes). The reported results (averaged over 10 runs) show that even for the densest and most redundant case ($N = 8{,}000,\ k = 100$), the homotopy is not well captured (n_classes = 3.2/4) by the k-nearest PRM. Moreover, the large size of the computed roadmap results in a significant computing time (3,819 seconds) owing to the amount of collision tests required to add new nodes and edges. Comparatively, our method captures the four homotopy classes in only 37 sec-

Table 1. General performance for the construction of roadmaps.

Environments	2D Vis	2D first-order	2D second-order	Laby	Indoor	Square	Helico	Arm
Figure	18(a)	18(b)	18(c)	19(a)	19(b)	20	21(a)	21(b)
DOF	2	2	2	2	2	3	6	6
Nodes	20	71	44	149	66	12	30	41
Edges	19	121	34	177	83	14	39	46
Cycles	0	51	14	29	18	3	10	6
Time (s)	2	8	16	109	164	37	56	99

Table 2. Time repartition for the constructions of roadmaps (as a percentage).

Environments	2D Vis	2D first-order	2D second-order	Laby	Indoor	Square	Helico	Arm
Vis-PRM	100	19	13	19	25	24	5	12
Subroadmap	–	75	15	32	20	61	9	70
Redundancy	–	–	66	35	49	11	80	13
Other	0	6	6	14	6	4	6	5

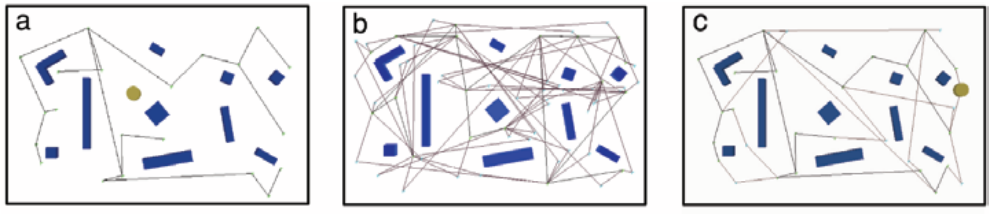


Fig. 18. Comparison between the three algorithms of roadmap construction: (a) Visibility-PRM; (b) first-order and (c) second-order deformation roadmap.

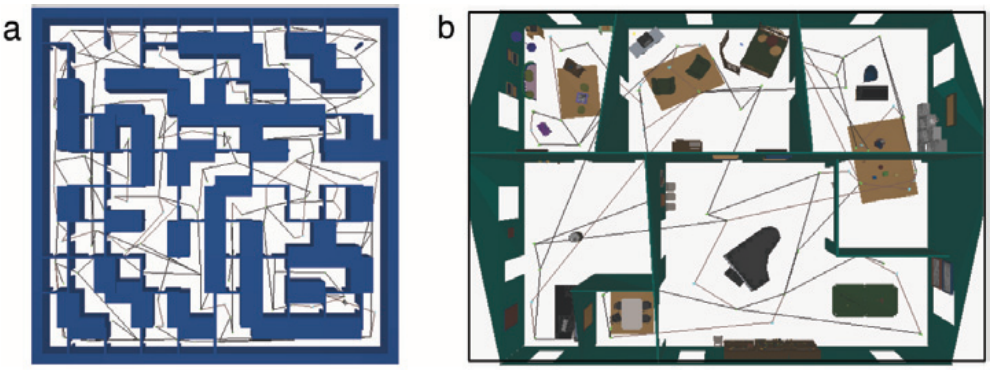


Fig. 19. PDRs for two-dimensional environments: (a) a labyrinth with many homotopy classes; (b) an indoor environment with complex geometry.

onds. The high speed-up comes from the very compact size of the PDR (only 12 nodes) which largely compensates for the additional cost of filtering the useless redundant cycles.

The last set of experiments (Figure 21) involves 6-DOF robots in three-dimensional environments. In the first case (free-flying robot), the free space has only one homotopy class.

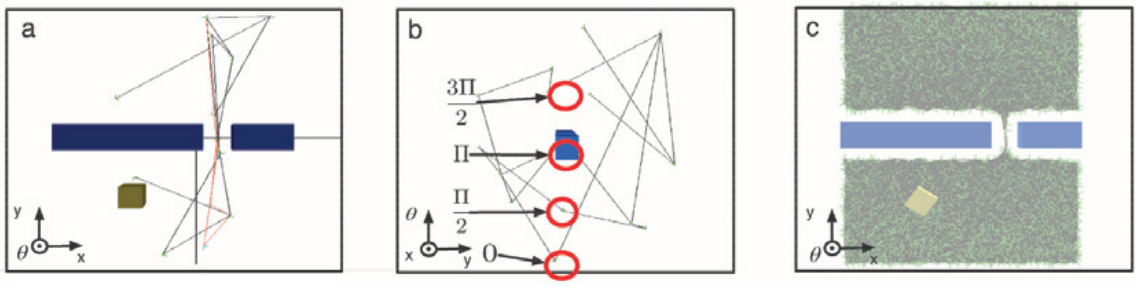


Fig. 20. PDR capturing the four homotopy classes for a rotating square and a narrow passage: (a) (x, y) view of the deformation roadmap, (b) (y, θ) view of the same roadmap showing the four kinds of passage found in $\mathcal{C}$; (c) comparison with the dense roadmap obtained with a classic k-nearest PRM.

Table 3. Homotopy classes found by a k-nearest PRM for the problem described in Figure 17.

		n_classes			Time (s)		
	N	$k = 10$	$k = 20$	$k = 100$	$k = 10$	$k = 20$	$k = 100$
k-	1,000	0.1	0.2	1.2	6.4	9.3	33.2
nearest	2,000	0.1	0.6	1.6	33.2	43.5	110.0
PRM	4,000	0.8	1.0	2.8	246	336	455
	8,000	1.4	2.4	**3.2**	2,947	3,295	**3,819**
PDRoadmap	12		**4**			**37**	

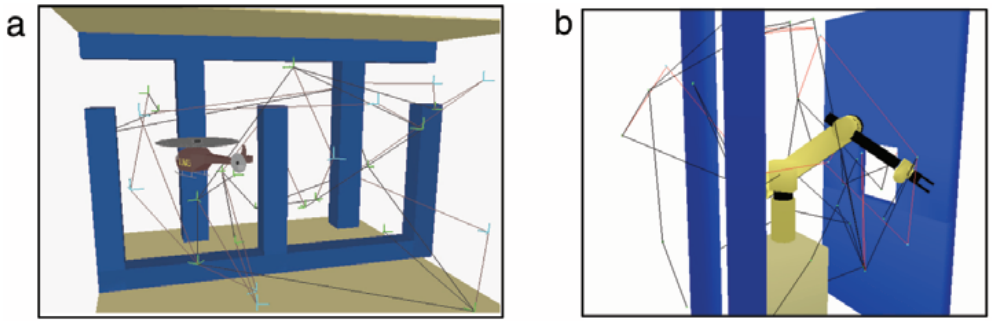


Fig. 21. PDRs for three-dimensional environments: (a) free-flying robot; (b) 6-DOF manipulator arm.

Thus, a roadmap based on homotopy would have a tree structure. The results show that our method makes it possible to build a compact roadmap (in 56 seconds) while capturing a richer variety of paths than the homotopy. The second scene concerns a 6-DOF manipulator arm where six nodes (and 12 edges) are added to the visibility roadmap (total time of 99 seconds) to represent the complexity of the space.

Table 1 summarizes the performance results and Table 2 provides a break-up of the total computational effort by showing the respective contributions of the visibility tree building and the cycle addition stages.

7. Conclusion

We have presented a general method to build compact PDRs with useful cycles representative of the different varieties of free paths of the configuration space. The introduction to these cycles is important to obtain higher quality solutions when postprocessing queries inside the roadmap. Our approach is based on the notion of path deformability indicating whether or not a given path can be easily deformed into another. Our experiments show that the method enables small roadmaps to reliably capture the multiple connectedness of possible com-

plex configuration spaces. Several improvements remain for further work. First of all, the method has been tested so far for free-flying and articulated robots with up to 6 DOFs. Further evaluation of its performance is needed for higher DOF articulated robots. Moreover, we would like to further investigate the link between the varieties of free paths stored in the roadmap and the smoothing method used to shorten the solution paths when postprocessing queries. Finally, another improvement concerns the extension to robots with kinematically constrained motions (e.g. nonholomic or closed chain robots) requiring the use of a non-linear local method.

Acknowledgment

This work was supported by the European projects MOVIE IST-20001-39250 and PHRIENDS IST-045359.

References

Amato, N. M., Bayazit, O. B., Dale, L. K., Jones, C. and Vallejo, D. (1998). OBPRM: an obstacle-based PRM for 3D workspaces. *Robotics: The Algorithmic Perspective (WAFR1998)*, Agarwal, P., Kavraki, L. E. and Mason, M. (eds). Wellesley, MA, A.K. Peters, pp. 155–168.

Bohlin, R. and Kavraki, L. E. (2000). Path planning using lazy PRM. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 521–528.

Boor, V., Overmars, M. H. and van der Stappen, A. F. (1999). The Gaussian sampling strategy for probabilistic roadmap planners. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1018–1023.

Burns, B. and Brock, O. (2005). Toward optimal configuration space sampling. *Proceedings of Robotics: Science and Systems*.

Cheng, H.-L., Hsu, D., Latombe, J.-C. and Sánchez-Ante, G. (2006). Multi-level free-space dilation for sampling narrow passages in PRM planning. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1255–1260.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E. and Thrun, S. (2005). *Principles of Robot Motion*. Cambridge, MA, MIT Press.

Geraerts, R. and Overmars, M. H. (2006). Creating high-quality roadmaps for motion planning in virtual environments. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Hatcher, A. (2002). *Algebraic Topology*. Cambridge, Cambridge University Press. Available at http://www.math.cornell.edu/∼hatcher/AT/ATpage.html.

Holleman, C. and Kavraki, L. E. (2000). A framework for using the workspace medial axis in PRM planners. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1408–1413.

Hsu, D., Jiang, T., Reif, J. and Sun, Z. (2003). The bridge test for sampling narrow passages with probabilistic roadmap planners. *Proceedings IEEE International Conference on Robotics and Automation*.

Huang, Y. and Gupta, K. (2004). A Delaunay triangulation based node connection strategy for probabilistic roadmap planners. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 908–913.

Jaillet, L. (2005). Méthodes Probabilistes Pour La Planification Réactive de Mouvements. *PhD Thesis*, Paul Sabatier University. Available at http://robotics.cs.umass.edu/tc-apc/Main/Theses.

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

Koenig, S. and Furcy, D. (2004). Lifelong planning A*. *Artificial Intelligence*, **155**: 93–146.

Kurniawati, H. and Hsu, D. (2006). Workspace-based connectivity Oracle: an adaptive sampling strategy for PRM planning. *Algorithmic Foundations of Robotics VII*, Akella, S. et al. (eds). Berlin, Springer.

Latombe, J.-C. (1991). *Robot Motion Planning*. Dordrecht, Kluwer.

LaValle, S. M. (2004). *Planning Algorithms*. Cambridge, Cambridge University Press. Available at http://msl.cs.uiuc.edu/planning/.

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

Nieuwenhuisen, D. and Overmars, M. H. (2004). Useful cycles in probabilistic roadmap graphs. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 446–452.

Rodriguez, S., Shawna, S., Pearce, R. and Amato, N. M. (2006). Resampl: a region-sensitive adaptive motion planner. *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*.

Saha, M. and Latombe, J.C. (2005). Finding narrow passages with probabilistic roadmaps: the small-step retraction method. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Schmitzberger, E., Bouchet, J. L., Dufaut, M., Didier, W. and Husson, R. (2002). Capture of homotopy classes with probabilistic road map. *Proceedings of the IEEE/RSJ International Conference on Robots and Systems*.

Sekhavat, S., Svestka, P., Laumond, J.-P. and Overmars, M. H. (1998). Multi-level path planning for nonholonomic robots using semi-holonomic subsystems. *The International Journal of Robotics Research*, **17**(8): 840–857.

Siméon, T., Laumond, J.-P. and Nissoux, C. (2000). Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics Journal*, **14**(6): 477–494.

Siméon, T., Laumond, J.-P. and Lamiraux, F. (2001). Move3D: a generic platform for path planning. *Proceedings IEEE International Symposium on Assembly and Task Planning*.

Sánchez, G. and Latombe, J.-C. (2002). On delaying collision checking in PRM planning—application to multi-robot coordination. *The International Journal of Robotics Research*, **21**(1): 5–26.

van den Berg, J. P., Nieuwenhuisen, D., Jaillet, L. and Overmars, M. H. (2005). Creating robust toadmaps for motion planning in changing environments. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

van den Berg, J. P. and Overmars, M. H. (2007). Kinodynamic motion planning on roadmaps in dynamic environments. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Wilmarth, S., Amato, N. M. and Stiller, P. (1999). MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the free space. *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1024–1031.