Computer-Aided Design 75–76 (2016) 47–60

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Path planning with obstacle avoidance by G^1 PH quintic splines

Carlotta Giannelli [a,*], Duccio Mugnaini [b], Alessandra Sestini [a]

[a] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Viale Morgagni 67/A, I-50134 Firenze, Italy

[b] Dipartimento di Scienze e Alta Tecnologia, Università degli Studi dell'Insubria, Via Valleggio 11, I-22100 Como, Italy

ARTICLE INFO

Article history:
Received 12 July 2015
Accepted 5 February 2016

Keywords:
Path planning
Obstacle avoidance
Spline
Pythagorean-hodograph curves
Tension parameters

ABSTRACT

We propose a two-step approach for the construction of planar smooth collision-free navigation paths. Obstacle avoidance techniques that rely on classical data structures are initially considered for the identification of piecewise linear paths having no intersection with the obstacles of a given scenario. Variations of the shortest piecewise linear path with angle-based criteria are proposed and discussed. In the second part of the scheme we rely on spline interpolation algorithms with tension parameters to provide a smooth planar control strategy. In particular, we consider the class of curves with Pythagorean structures, because they provide an exact computation of fundamental geometric quantities. A selection of test cases demonstrates the quality of the new motion planning scheme.

1. Introduction

The design of motion planning strategies plays a fundamental role in modern computer applications with focus on different kinds of simulation environments naturally related to robotics, as well as to scientific visualization and interactive navigation [1,2]. The issue of finding an optimal trajectory for a given path should properly combine the geometric part of the motion, usually identified by a *path planning scheme*, with a suitable time law.

The path planning problem includes the identification of paths that do not intersect any obstacle. In order to avoid forbidden configurations related to a given scenario, several graph-like structures may be considered, see for example [3] for a recent survey related to possible collision-free piecewise linear solutions. Using a standard graph search algorithm, a graph with non-negative edge weights can be exploited to compute the path with *lowest total cost* between any two vertices of the graph. In particular, the output of the algorithm may return an optimal path with respect to a *distance* (shortest path) criterion. In order to provide an optimal trade-off between the accuracy of the prescribed trajectory and the flexibility of interactive navigations, the information concerning the collision-free piecewise linear path may be subsequently combined with spline interpolation techniques that provide a smooth planar control strategy, see e.g., [4]. Previous attempts in this direction usually considered solutions related to classical spline methods [5,6].

By considering interpolation schemes with tension control – see, e.g., [7,8] and the references therein – as a control tool on the shape of the interpolating curve, we present a two-step approach for *smooth* path planning with obstacle avoidance. In the first step, algorithms for the modification of the shortest piecewise linear path associated to the trapezoidal map and the visibility graph according to simple angle-based criteria are proposed and discussed. In the second step, we consider the class of *curves with Pythagorean structures*, because they usually provide paths with fair shape and always guarantee exact computation of fundamental geometric quantities like curvature and arc length [9]. This can also facilitate the physical part of the motion which requires accurate arc length and curvature computations [10].

The structure of the paper is as follows. Section 2 provides the preliminary material that introduces the problem setting and the two graph structures considered in the subsequent algorithms, namely the trapezoidal map and the visibility graph. The design of a piecewise linear collision-free path is addressed in Section 3. Different algorithms that rely on the information provided by the above mentioned data structures are presented and discussed. Section 4 provides a brief overview of Pythagorean-hodograph (PH) curves by focusing on G^1 PH quintic Hermite spline interpolants with tension parameters. An asymptotic analysis that can be exploited for choosing the free parameters involved in the interpolation scheme is developed in Section 5. A final illustrative example in a non-trivial obstructed scenario is presented in Section 6. Finally, Section 7 concludes the paper.

* Corresponding author.
E-mail addresses: carlotta.giannelli@unifi.it (C. Giannelli), dmugnaini@uninsubria.it (D. Mugnaini), alessandra.sestini@unifi.it (A. Sestini).

http://dx.doi.org/10.1016/j.cad.2016.02.004
0010-4485/

2. Data structures

The aim of *path planning* is to define a suitable collision-free path from an initial to a target position in a certain scenario that includes a set of obstacles. In robotics, the position of every point of the robot is usually defined as its *configuration* $\mathbf{q}$ and depends on the specific robot model. The parameters needed to specify the robot configuration identify the *degrees of freedom* associated to the problem. All possible configurations define the *configuration space* $\mathcal{Q}$ (*C-space*) that can be divided in two subsets: $\mathcal{Q}_{free}$, the set of robot configurations free from collision with obstacles, and $\mathcal{Q}_{obst}$, the obstacle configuration space given by the set of intersection configurations between the robot and any obstacle. Hence, a configuration $\mathbf{q}$ is just a point (or element) in the configuration space and the number of the degrees of freedom corresponds to the dimension of $\mathcal{Q}$ [1].

Given a two- or three-dimensional rigid body and a certain scenario with a prescribed set of obstacles, we look for a smooth vector function $\mathbf{r}(t) : [0, 1] \to \mathcal{Q}$ so that

$$\mathbf{r}(0) = \mathbf{q}_s, \qquad \mathbf{r}(1) = \mathbf{q}_g, \quad \text{and} \quad \mathbf{r}(t) \in \mathcal{Q}_{free} \quad \forall t \in [0, 1],$$

where $\mathbf{q}_s$ and $\mathbf{q}_g$ correspond to the start and goal configuration, respectively. The desired path is the image in $\mathcal{Q}$ of $\mathbf{r}(t)$.

The path planning problem can be considered in a wide set of scenarios by taking into account different problem settings. In particular, we consider the simplest planar *point robot problem* where a circular robot of radius r is moving within a certain scenario characterized by convex obstacles of polygonal shapes. Consequently, the configuration space is just obtained by augmenting the obstacle boundaries by their polygonal offset at distance r. Then, after this preliminary offset computation, we can consider the robot as a moving point within a two-dimensional environment characterized by a known set of static obstacles with a fixed position. In this context, the robot follows an a priori identified path, constructed according to some *off-line* controller.

In order to identify a feasible path, certain *navigability structures* that contain information about the free configuration space are usually exploited. Several techniques to obtain different navigability structures are available. Among others, two approaches that provide alternative solutions based on suitable data structures to navigate $\mathcal{Q}_{free}$ are the exact cell decomposition and roadmap methods. In the family of graphs that may be taken into account within these methods, see e.g., [1,11], we consider the *trapezoidal map* and the *visibility graph*.

The *trapezoidal map* (or trapezoidal/vertical decomposition) is a well-known example of exact cell decomposition. Given a polygonal environment, the first step is to define a bounding box that includes all obstacles. A trapezoidal map is conventionally obtained by drawing two vertical extensions (one going upwards and the other going downwards) from every vertex of the obstacles to the first intersection with an obstacle edge or to the bounding box. Once Q_{free} is partitioned, the adjacency graph is defined by placing a node inside each trapezoid, e.g. its geometric centroid, and additional ones in the middle of the vertical extensions. An arc is then defined between two vertices of these two kinds of nodes associated to the same trapezoid. In order to construct the target path, $\mathbf{q}_s$ and $\mathbf{q}_g$ are added to the graph by simply connecting them to the central vertices $\mathbf{v}_s$ and $\mathbf{v}_g$ of the two corresponding trapezoids. Different graph search algorithms on the trapezoidal map identify admissible collision-free paths between $\mathbf{v}_s$ and $\mathbf{v}_g$. An example of trapezoidal map is shown in Fig. 1(top).

The *visibility graph* is a widely known roadmap method in computational geometry. Given a set of polygonal obstacles in the plane, the nodes of the visibility graph correspond to the vertices of each polygon. An arc between two vertices belongs to the graph if the linear segment that connects these vertices

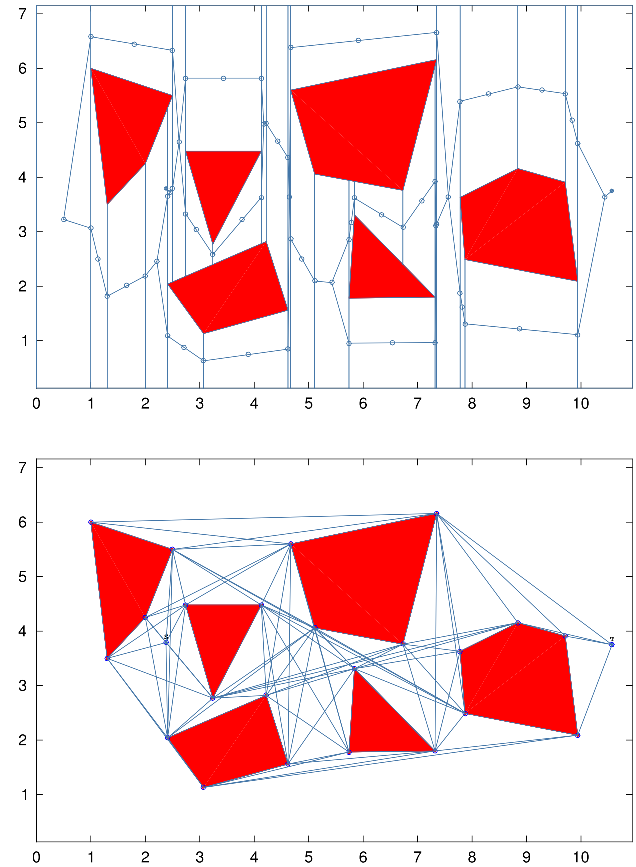

Fig. 1. A simple scenario (S1) together with the corresponding trapezoidal decomposition (top) and visibility graph (bottom).

does not intersect any obstacle. To complete the roadmap, the start and target positions are added to the set of graph nodes together with the corresponding arcs. The naive algorithm to compute the visibility graph has complexity $O(n^3)$, where n is the number of nodes in the graph. By considering a suitable sweep approach with balanced search tree structures the complexity can be reduced to $O(n^2 \log n)$ [11]. Fig. 1 (bottom) shows the visibility graph associated to a simple scenario. Other examples of roadmap methods include *Voronoi diagrams* and *silhouette graphs*.

3. Angle based algorithms for collision-free piecewise linear paths

In order to define a smooth path that does not intersect any obstacle, we consider two consecutive steps. In the first phase, we rely on one of the two data structures introduced in the previous section to define a suitable collision-free piecewise linear path. Subsequently, the smooth path is obtained by interpolating the vertices of the piecewise linear path previously computed, as described in Section 4.

Classical graph search algorithms can be used either on the trapezoidal map or the visibility graph to identify admissible paths that do not intersect any obstacles in the first step of the method. A standard choice relies on the Dijkstra's algorithm that computes the *shortest path* between two vertices of a given graph [12]. Note that the shortest path associated to the visibility graph is the *absolute* shortest path from the start to the goal position with respect to the considered scenario, see e.g., [11].

In order to obtain a final path defined by a curve without significant curvature peaks, the identification of suitable piecewise linear paths with a small angle between two adjacent segments