



**HAL**  
open science

## Multi-layer path planning control for the simulation of manipulation tasks: involving semantics and topology

Simon Cailhol, Philippe Fillatreau, Yingshen Zhao, Jean-Yves Fourquet

### ► To cite this version:

Simon Cailhol, Philippe Fillatreau, Yingshen Zhao, Jean-Yves Fourquet. Multi-layer path planning control for the simulation of manipulation tasks: involving semantics and topology. *Robotics and Computer-Integrated Manufacturing*, 2019, 57, pp.17-28. 10.1016/j.rcim.2018.10.010 . hal-02111780

**HAL Id: hal-02111780**

**<https://hal.science/hal-02111780v1>**

Submitted on 26 Apr 2019

**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.







## Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/21730>

### To cite this version:

Cailhol, Simon  and Fillatreau, Philippe  and Zhao, Yingshen   
and Fourquet, Jean-Yves  *Multi-layer path planning control for the simulation of manipulation tasks : involving semantics and topology.* (2019) *Robotics and Computer-Integrated Manufacturing*, 57. 17-28. ISSN 0736-5845

Any correspondence concerning this service should be sent to the repository administrator: [tech-oatao@listes-diff.inp-toulouse.fr](mailto:tech-oatao@listes-diff.inp-toulouse.fr)

# Multi-layer path planning control for the simulation of manipulation tasks: Involving semantics and topology

Simon Cailhol, Philippe Fillatreau\*, Yingshen Zhao, Jean-Yves Fourquet

Laboratoire Genie de Production (LGP), Université à Fédérale de Toulouse, INPT/ENIT, 47 avenue d'Azereix, 65000 Tarbes, France

## ABSTRACT

*Keywords:*  
Path planning  
Environment modeling  
Geometry  
Topology  
Semantics

The industrial and research communities show increasing interest in using automatic path planning techniques for the simulation of manipulation tasks. Automatic path planning, largely explored by the robotics community over the past 30 years, computes the trajectories of robots or manipulated parts. However, as techniques developed so far use mostly purely (and large) geometric models, they may fail, produce a trajectory of little relevance, or lead to very high computation times, when facing complex or very constrained environments. Involving higher abstraction level information should lead to better relevance of the simulation. In this paper, we propose a novel path planning technique relying on an original multi-layer environment model containing geometrical, topological and semantic layers. A first coarse planning step at the topological and semantic layers and a fine planning step at the local and semantically characterized geometrical layer form the path planning process. Experimental full-scale results show increased control on the planning process, leading to much lower computation times and increased relevance of the computed trajectory .

## 1. Context and motivation

While industrial companies reduce time and cost of product development, manufactured products are increasingly complex and integrated. In order to face these challenges, there is an increasingly strong need for the use of integrated numerical tools in all the stages of the industrial Product Lifecycle Management. In particular, virtual prototypes should provide a way to validate, through simulation and from design stage on, the actions (e.g., assembly or maintenance) related to the lifecycle of the product. This requires computing trajectories for objects or parts in motion in digital environments (typically CAD models) potentially strongly geometrically constrained.

In this work, we are interested in using automatic path planning techniques for the simulation of manipulation tasks performed by a human operator. The robotics community has been developing automatic path planning algorithms since the 80's. These algorithms mainly rely on the use of dense purely geometrical models (e.g. surfaces meshes of both the moving part and the obstacles in the environment). In large and highly geometrically constrained environment models, traditional automatic path planning techniques may fail, lead to high computation times or propose trajectories of little relevance to the task to be simulated.

However, a human operator generally finds solutions to complex

assembly problems in a reasonable time, as he involves his cognitive capacities and knowledge of the task to be performed. His understanding of the environment and of the manipulated objects is not limited to their geometry but may involve higher abstraction level information.

Therefore recent works proposed interactive path planning approaches where an automatic path planner and a human being cooperate. As the automatic planners still use purely geometric models, interaction is of little relevance to the task to be performed and real-time constraints are often out of reach. Higher abstraction level data should be considered. Indeed, the scientific community has started to explore the interest of considering topological or semantic information for knowledge representation, human-robot interaction, or mobile robots navigation.

Our goal here is to improve the performances of automatic path planners for the simulations of complex tasks performed by a human operator, or the assistance to manipulation.

In this paper, we propose a novel path planning strategy relying on the use of higher abstraction level information than the purely geometrical data traditionally used

1. A multi-layer environment model containing geometrical, topological and semantic data;

\* Corresponding author.

E-mail address: [philippe.fillatreau@enit.fr](mailto:philippe.fillatreau@enit.fr) (P. Fillatreau).

## 2. A path planning strategy involving these heterogeneous environment data.

The adopted approach allows a better control of the path planning algorithms, leading to significantly lower computation times and more relevant proposed paths. We demonstrate these benefits over state-of-the-art automatic path planning for a full scale complex manipulation task.

The paper is organized as follows: in [Section 2](#), the state of the art of automatic path planning techniques is discussed. Then, [Section 3](#) introduces our novel environment model involving geometrical, topological and semantic layers. [Section 4](#) presents a novel path planning architecture allowing to take advantage of the heterogeneous information available in the environment model. [Section 5](#) details the implementation of both the environment model and the automatic path planner. [Section 6](#) provides experimental results on a complex full scale manipulation task and presents the benefits of our approach over state of the art methods. [Section 7](#) concludes and gives perspectives for this work.

## 2. State of the art

Since the pioneering work by Lozano-Perez et al. [1] in the 80s, path planning has been a fertile research field. The purpose is to find a path from a start configuration ( $S$ ) to a goal configuration ( $G$ ) for a moving or manipulated object in the *free-space* (i.e., a path with no collision with the obstacles of the environment). We propose to classify the path planning techniques of the literature according to the environment models used and the way to explore the environment. First, considering the environment models involved, we distinguish global approaches (where a map of the whole environment is used) and local approaches (where local models of the neighborhood of the moving object are used). Then, we also distinguish the techniques involving a deterministic exploration of the environment and those performing a probabilistic one.

The techniques involving a global approach and a deterministic exploration of the environment use a full or *exact* (up to the resolution of the model) (purely) geometric model or cartography of the free-space of the environment [2]. A deterministic exploration of the cartography is then performed to find the desired trajectory. In the case of a static environment, the model may be computed off-line and reused for many path planning queries. The strong advantage of these techniques is their completeness. Their main drawback is that the time needed to compute the models of the free-space becomes prohibitive for problems with more than three dimensions [3].

The techniques involving a local approach and a deterministic exploration use potential fields (attractive potential for the goal and repulsive potential for obstacles) [4]. As opposed to the global approaches presented formerly, such local approaches lead to lower computational times. However, they are not complete and thus may fail at finding an existing path.

The techniques involving a global approach and a probabilistic exploration of the environment usually involve the building of a global roadmap connecting configurations randomly drawn. A given set of configurations is first drawn, then these configurations are connected in a roadmap. Once the roadmap has been built, the start and goal configurations are connected to the roadmap which is then explored to find the desired path [5]. The main issue here is to define the number of random configurations needed to build a relevant roadmap of the whole environment.

Finally, techniques involving a local approach and a probabilistic exploration of the environment usually iteratively extend the roadmap, from given configurations (start, both start and goal, or other configurations) e.g. RRT Rapidly-exploring Random Tree [6] or RDT Rapidly-exploring Dense Tree). Newly randomly drawn configurations are integrated locally to the roadmap under construction until the start and

goal configurations are connected.

Probabilistic techniques with a global or a local approach are probabilistically complete. However, they do not guarantee finite-time solutions.

However, a human operator generally performs the corresponding tasks in a more relevant way, and finds solutions to the assembly problem set in a reasonable time. This is due to the fact that the operator involves his cognitive capacities and trade-oriented knowledge of the task to be performed. His understanding of the environment and of the manipulated objects is not limited to their geometry but may involve higher abstraction level information such as their functionalities or topological properties. Similarly, humans use two kinds of path planning strategies: (a) first, a coarse planning step using a static representation of the environment derived from scenes learned before (b) second, a fine path planning step used for dynamic mapping and fine tuning of the rough path [7].

In order to take advantage of the cognitive capacities of the operator in the simulation of manipulation tasks, some works have started to explore interactive path planning approaches with an operator in the loop cooperating with the automatic path planner [8–11]. Such approaches require to consider the issues of control sharing between the planner and the human operator [12]. Due to the limitations of automatic path planning addressed above, processing times may be too large for real-time interaction, while the path planner may explore solutions of little relevance for the operator.

Higher abstraction level data should be considered. Indeed, the scientific community has started to explore the interest of considering topological or semantic information for environment modeling [13–18], robot localization [19,20], navigation [21], task planning [22–25] or grasping [26–28].

Our goal here is to improve the performances of path planners for the automatic or interactive simulations of complex tasks performed by a human operator.

To overcome path planning techniques limitations due to the heavy purely geometrical model of the environment used (computation time, failures, poor relevance), our idea is to develop a general framework based on digital environment models involving higher abstraction level data than the traditionally used geometrical models. Two elements compound our general framework:

1. A three-layers environment model provides the classical geometrical information but also topological and semantic information about the rigid bodies and the free-space. This abstract information can model any useful knowledge for the tasks under consideration.
2. An automatic path planner performs a two-step path planning. The first step called coarse planning defines a topological path within the topology of the environment. The topological path is split to form topological steps. Then, the second step called fine planning defines a geometrical trajectory for each topological step. The semantic information contained in the environment model is used to control both planning steps. First, this semantic information is used to guide the exploration of the topological model of environment during the coarse planning. Then, the planner chooses relevant geometrical path planning techniques to be used for each topological step according to the local semantic information in the environment model.

Our novel multi-layer path planning approach leads to:

- Improvement of the path planning process qualitative performance (relevance of paths),
- Improvement of the path planning process quantitative performance (success rates and computation load).

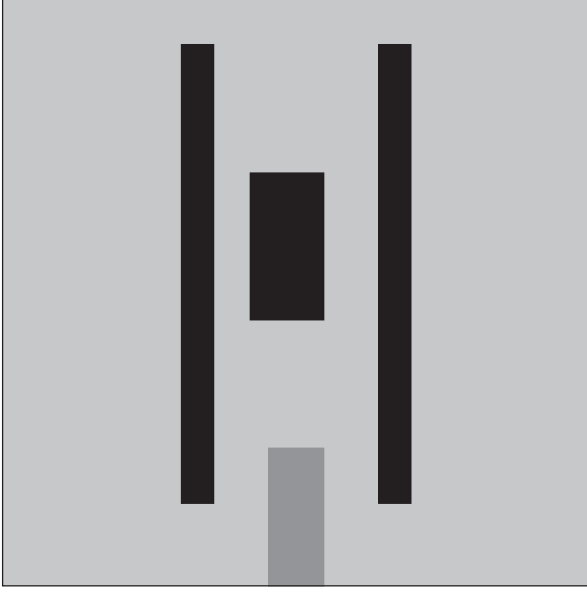


Fig. 1. Illustration case environment.

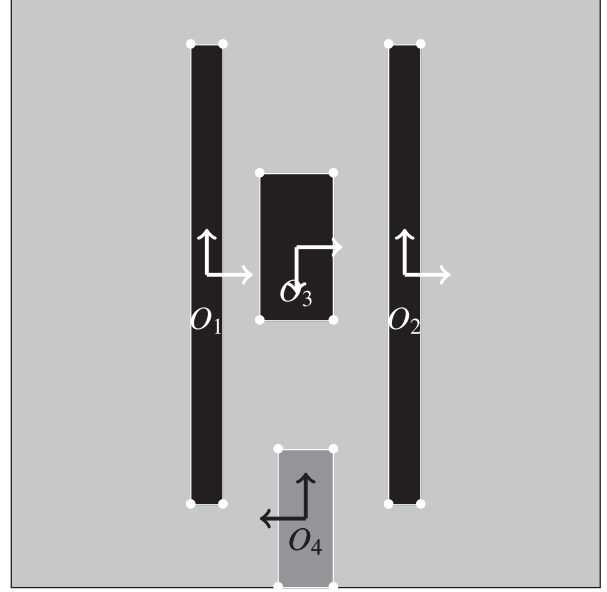


Fig. 2. Rigid bodies geometrical model.

### 3. Environment model

The environment is considered as a closed part of the Cartesian space cluttered with static and/or mobile obstacles considered as rigid bodies. The proposed model of the environment is made of two parts: a rigid bodies model, and a free-space model. Both of these model involve different layers based on geometrical, topological or semantic information respectively.

In the simulations considered in this paper, rigid bodies do not move. However, we want our environment model to be compatible with future scenarios in which obstacles can move in the environment, or in which objects in the environment may be displaced to perform a task. Therefore, we define two kinds of rigid bodies in our environment model : the ones that may move or be displaced, labeled as “mobile”, and the ones that cannot move, labeled as “static”.

Fig. 1 depicts a 2D illustration case environment that will be used to introduce all the concepts involved in the proposed multi-layer environment model. It will also be used in Section 4 to describe the novel path planning process. The proposed concepts are illustrated with this 2D example but stand similarly for 3D cases.

The illustration case used (Fig. 1) is made of a square workspace cluttered with 4 rigid bodies where the gray one is mobile and the black ones are static.

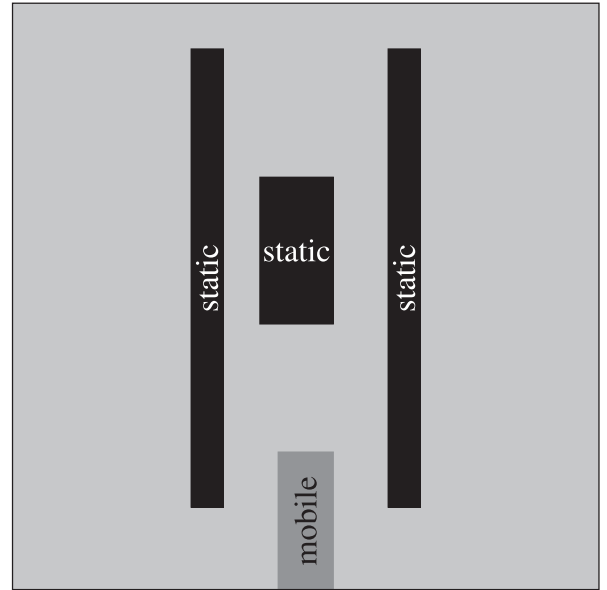


Fig. 3. Rigid bodies semantic model.

#### 3.1. Rigid bodies

The model of the rigid bodies is made of a geometrical and a semantic layers:

**Geometrical layer** (Fig. 2): the surfaces of the rigid bodies are represented using classical polyhedral models (vertices and edges displayed in white).

**Semantic layer** (Fig. 3): the semantic layer of the environment model allows defining attributes to describe high abstraction task- or trade-oriented knowledge. For example, the semantic attributes associated to a given rigid body may describe its shape, its function or the fact that it is static, or potentially mobile. These semantic information may be represented in various ways, e.g., using text, numerical values or a boolean. In our case, they are provided by the user but they could be obtained from inference systems.

#### 3.2. Free-space

The model of the free-space proposed is a global multi-layered map of the free-space of the environment.

**Geometrical layer** (Fig. 4): A cell decomposition based on an un-balanced tree (quadtrees in 2D or octrees in 3D) describes the geometry of the free-space [29]. This decomposition distinguishes three types of cells: the cells that do not intersect any rigid body (obstacles), the cells intersecting mobile rigid bodies only and the cells intersecting at least one static rigid body.

**Topological layer** (Fig. 5): The topological layer represents places (called  $P_i$ ) and borders (called  $B_{jk}$ ) and the topological relations between them (the border  $B_{jk}$  connects the two places  $P_j$  and  $P_k$ ). The model used is a topological graph in which the node  $n_{i,j}$  represents the borders  $B_{i,j}$  while the edges correspond to places. Each place or border is associated to a set of geometrical cells in the geometrical layer. The topological model of the environment is static. Mobile

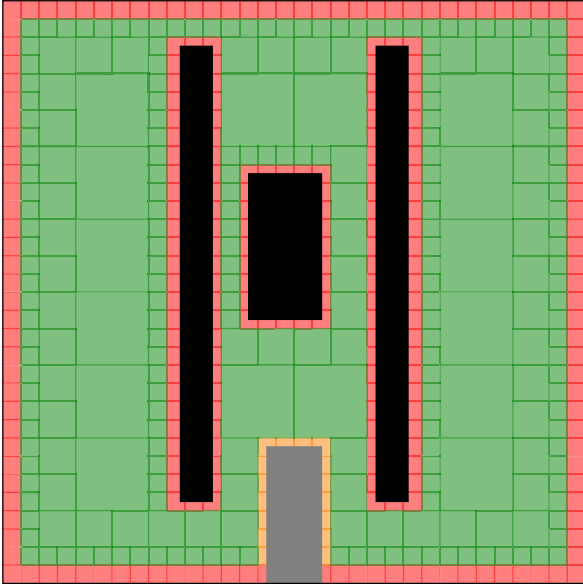


Fig. 4. Free-space geometrical model.

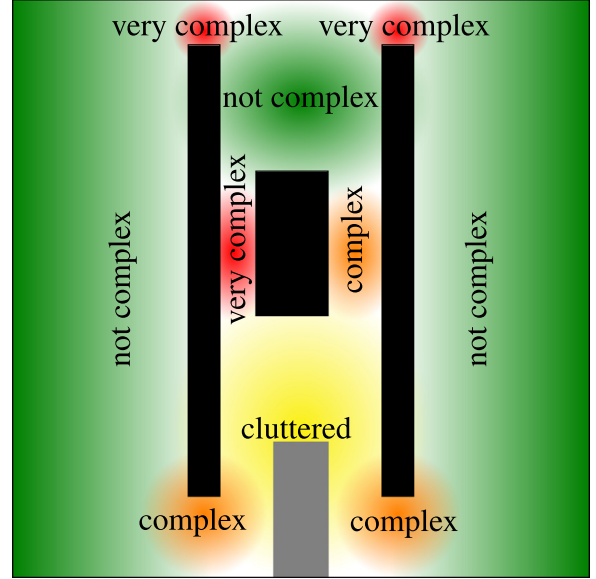


Fig. 6. Free-space semantic model.

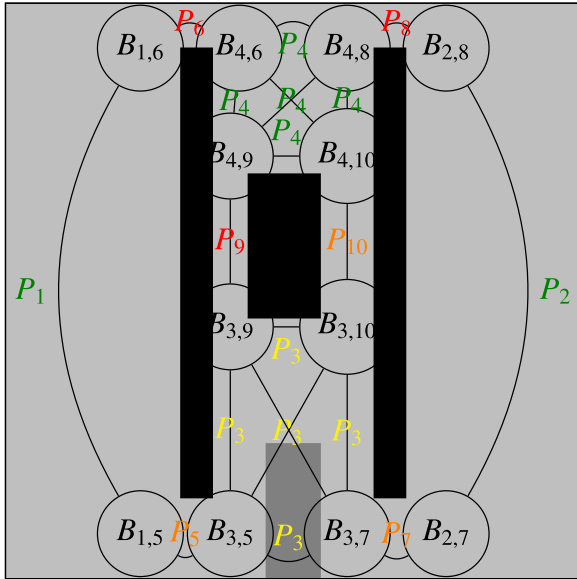


Fig. 5. Free-space topological model.

rigid bodies are not taken into account here.

**Semantic layer** (Fig. 6): The semantic layer is made of attributes attached to the places of the free-space. A given place is characterized by semantic information such as its *complexity* to be crossed because of its narrowness (in Fig. 6 places are represented in red, orange, or green, depending on the navigation complexity level) or its *cluttering* by mobile rigid bodies (yellow place in Fig. 6). Attributes may also be associated to describe different properties of a single place (for example place  $P_3$  is both *cluttered* and *not complex*).

### 3.3. Synthesis

The model of the environment represents both the rigid bodies and the free-space. It involves three layers modeling information of different abstraction levels: geometrical, topological and semantic. The geometrical layer uses classical models to represent both the rigid bodies and the free-space. The topological layer is a static representation of the places of the environment and their connectivity. Finally, the

semantic layer characterizes both the rigid bodies and the free-space with any useful information for the task to be performed. In the next section, we present a novel path planning approach using not only the geometrical but also the (higher abstraction level) topological and semantic information for a better control on the path planning process.

## 4. Path planning process

Path planning aims at defining, for a manipulated rigid body, a feasible geometrical trajectory between a start configuration and a goal configuration within the environment.

We propose an original two step path planning process to take advantage of the multi-layer environment model introduced in the previous section. The two steps are performed consecutively:

1. First a coarse planning step is performed. It defines a path within the topological graph. This topological path is then split into topological steps. Each step is made of a place to cross and a border to reach;
2. Then a fine planning step uses the geometrical model of both the rigid bodies and the free-space to define a geometrical trajectory for each topological step.

The proposed multi-layer path planning architecture involves several planners. Each of these planners uses a specific layer of the environment model:

- The semantic planner uses the semantic layer of the environment model to sequence the involvement of the other planners in the global path planning process (see Fig. 7).
- The topological planner explores the topological graph in order to compute a topological path (an ordered set of places and borders to cross).
- The geometrical planner explores the geometrical model (cell decomposition) of the free space, in order to find a path made of connected cells (typically by using a Dijkstra or A\* algorithm).
- The local planner actually defines a 6D trajectory (a set of 6D configurations defining the 3D position and the 3D orientation of the manipulated rigid body) solving a geometrical path planning query associated to a given topological step, this query being defined by the start and goal geometric milestones (configurations) defined during the coarse planning process (see Section 4.1). This planner is typically based on local path planning techniques such as RRT

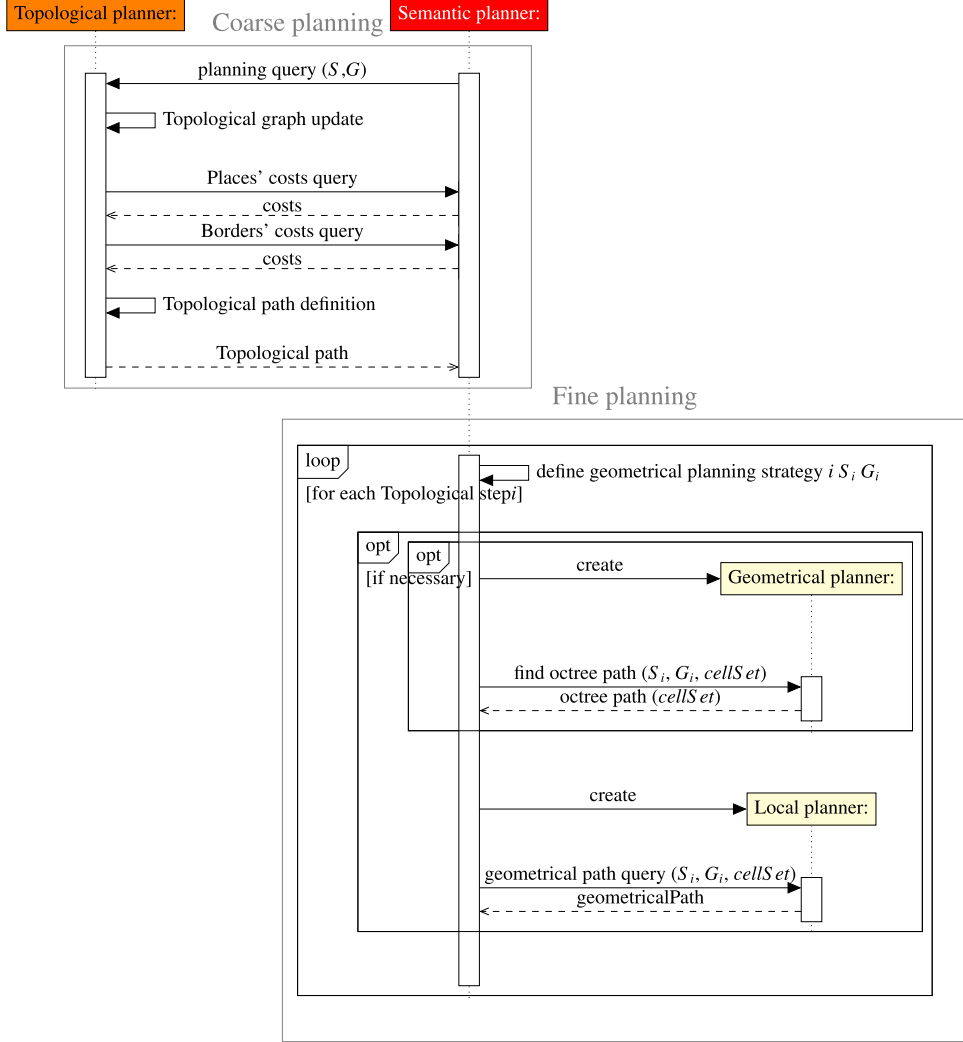


Fig. 7. Sequence diagram of the two-phases path planning process.

algorithms or potential fields.

Sections 4.1 and 4.2 detail how these planners interact during the path planning process, described in the sequence diagram given in Fig. 7, to define the trajectory, first at the topological level (coarse planning described in Section 4.1) and then at the geometrical level (fine planning described in Section 4.2).

#### 4.1. Coarse planning

The coarse planning step aims at finding a topological path linking  $S$  and  $G$ .

Fig. 8 illustrates the computation of a topological path. First, the topological graph is updated by adding two nodes corresponding to the start and goal configurations (Fig. 8). Then, the semantic planner controls the topological planning by defining the costs  $C_{n_{i,j}}$  and  $C_{e_k}$  of nodes and edges of the topological graph respectively. These costs are defined as functions of the semantic information associated to the places of the environment and to the manipulated object :

$$\begin{aligned}
 C_{n_{i,j}} &= f(\text{sem}(P_i), \text{sem}(P_j), \text{sem}(O_n)) \\
 C_{e_k} &= g(d_k, \text{sem}(P(e_k)), \text{sem}(O_h))
 \end{aligned}
 \quad (1)$$

where  $n_{i,j}$  is the node corresponding to border  $B_{i,j}$  connecting places  $P_i$  and  $P_j$  and  $e_k$  the edge associated to place  $P_k$ ; the  $\text{sem}()$  function provides the semantic information associated to a given element (place or

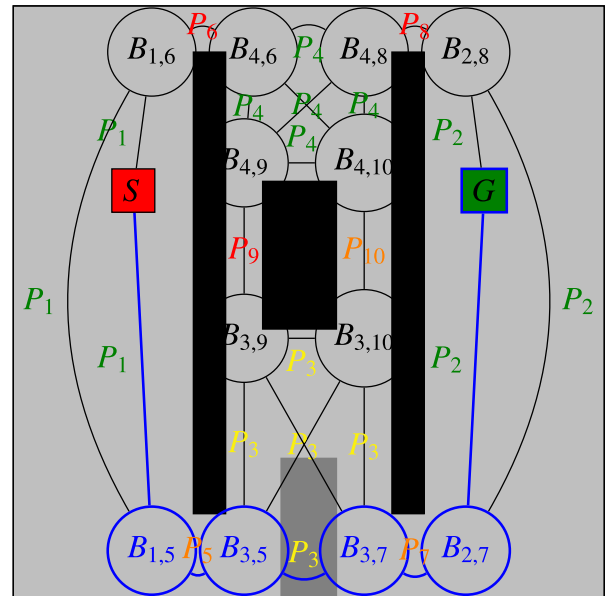


Fig. 8. Selected topological path (blue nodes and edges of topological graph).

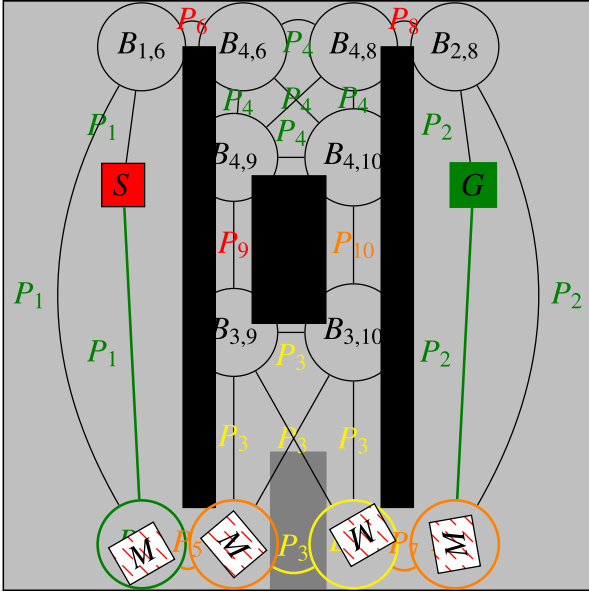


Fig. 9. Topological steps and red striped milestones defined during the coarse planning.

rigid body),  $O_h$  is the rigid body the trajectory is being planned for, and  $d_k$  is a geometrical distance associated to edge  $e_k$ . As the nodes  $S$  and  $G$  are not linked to borders but are necessarily in the topological graph, their cost is set to zero.

Eq.(1) shows that the costs of the topological graph depend on the rigid body  $O_h$  for which the trajectory is being planned and on the places to be crossed.

As shown in Fig. 7, once the semantic planner has defined the costs of the topological graph elements, the topological planner explores the topological graph (typically with a Dijkstra or A\* algorithm) and finds the path with the lowest cost (blue nodes and edges in Fig. 8).

The topological path obtained is then split into topological steps. Each topological step is made of a place to cross  $P_i$  and a border to reach  $B_{i,j}$  (defining that way borders as topological milestones). Finally, once the topological path and its steps have been defined, geometrical milestones (red striped configurations  $M$  in Fig. 9) are randomly drawn within each topological milestone (i.e. border), to be used as start and goal configurations for the path planning queries, involved at the fine planning phase (see Section 4.2), and associated to each topological step. The goal of the last topological step is  $G$ .

#### 4.2. Fine planning

The fine planning step is performed on each topological step in order to define a local geometrical trajectory (this process corresponds to the loop sequence given in Fig. 7). The geometrical milestones defined during the coarse planning step are used as start and goal configurations for geometrical planning queries defined for each topological step.

The semantic planner controls the fine planning phase by choosing the planning algorithms to adapt the planning strategy to the place to cross (according to its semantics). The proposed architecture potentially allows involving any existing path planning algorithm in the state of the art and choosing among them the algorithms most adapted to the local semantic context of the topological step involved. In the least complex cases (e.g., in the case of a free place, easy to cross), the 6D trajectory may be defined using a simple linear interpolation between the start and goal configurations; in more complex cases, the local planner may be used on its own; in the most complex cases, the 6D trajectory may be found by successively calling the geometrical planner, allowing to define a path of connected cells of the geometric model, and then the local

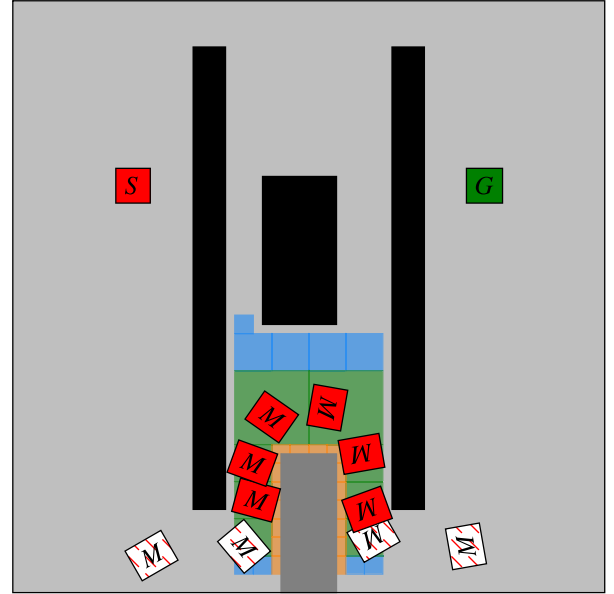


Fig. 10. Octree path and intermediate red milestones defined during the fine planning.

planner; in this last case, we reduce the computational complexity by reducing the part of the free space explored by the local planner to the path (set of connected cells) found by the geometrical planner. The geometrical and local planners being involved in conditional strategies, the corresponding blocks in Fig. 7 are labelled as “opt”.

To illustrate these concepts, let us take some examples in our 2D illustration case (Fig. 10). Most of the places of the environment are not cluttered. The geometrical trajectory for the topological steps crossing such places is defined by the linear interpolation between its start and goal configuration. As a consequence, it does not need to involve any further planner. However, as the place  $P_3$  is cluttered by the mobile rigid body  $O_4$ , the geometrical trajectory of the third topological step has to be refined. To do so the geometrical planner first explores the part of the quadtree associated to the place  $P_3$ . During this process, the geometrical planner finds a sequence of connected cells (green cells in Fig. 10) between the start and goal configurations of the planning query of the topological step. Then, the local planner defines the geometrical 6D trajectory (red milestones  $M$ ) by exploring only the area (sequence of geometric cells) provided by the geometrical planner (Fig. 10).

#### 4.3. Synthesis

The two steps of the proposed automatic path planner are controlled thanks to the high abstraction level information involved in our multi-layer environment model. This control aims at improving the path planning process performances from both a qualitative (relevance of paths) and quantitative (success rate and computation time) point of view. The coarse planning step defines a topological path within the environment. The semantic information is used to choose the topological path with higher relevance. When defining the actual geometrical trajectory during the second step of fine planning, different path planning algorithms can be used depending on the local semantic information associated with the place belonging to the topological step considered.

### 5. Implementation

The proposed multi-layer environment model and path planner have been developed in C++ and integrated to the 3D simulation software Virtools®. Fig. 11 depicts the Unified Modeling Language (UML) domain model of the proposed architecture. The colors correspond to the layers of the architecture:



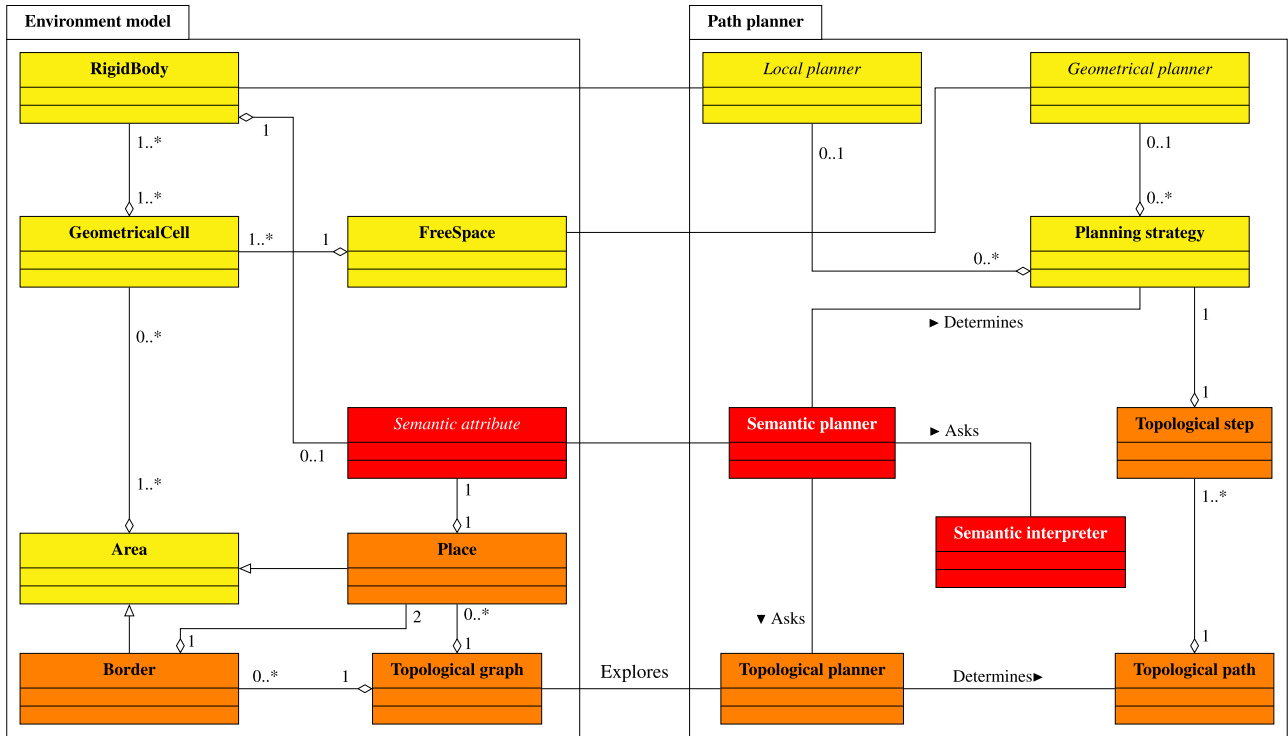


Fig. 11. Multi-layer environment and path planning UML domain model.

- **Geometrical** concepts are displayed in yellow,
- **Topological** concepts are displayed in orange,
- **Semantic** concepts are displayed in red.

### 5.1. Environment model

#### 5.1.1. Object oriented architecture

The geometrical model of the environment includes both the *Rigid bodies* and the *Free-space* models. The *Free-space* is described by the *Geometrical cells* of an unbalanced tree. The geometrical model also includes the concept of *Area* which consists in an aggregation of free *Geometrical cells*. On the topological layer, the *Area* is used to geometrically describe the topological *Places* and *Borders*. The connectivity of the *Places* and *Borders* is modeled by a *Topological graph*. Finally, *Semantic attributes* are attached to *Places* and *Rigid bodies*.

The modular structure of the *Semantic attribute* concept is implemented using an abstract class. The actual attributes are derived from this generic concept to allow representing various kinds of semantic information. As an example, the *Complexity* is described using an integer (the more complex the place is, the larger the integer is) while the *Cluttering* is a boolean. A textual attribute has also been implemented to describe the *Shape* of objects and places. The *Geometrical planner* and *Local planners* are also implemented in a modular way. They are implemented as generic classes which can be adapted to integrate solvers using potentially any path planning algorithm of the state of the art.

#### 5.1.2. Model building

To implement our original multi-layer environment model, well-known processes from the state of the art have been adapted, and others have been created. As an example, at the geometrical layer, the classical octree decomposition and its building process [30] have been adapted to specifically take into account the properties of rigid bodies (static or mobile).

The **octree decomposition** classically used in the literature [30] has been adapted to take into account the static and mobile objects (to

be evolutive towards dynamic scenarios, as explained at the beginning of Section 3). Indeed, all cluttered octree cells are associated with two lists of rigid bodies cluttering them. The first one aggregates the static rigid bodies, the second the mobile ones. The octree decomposition can be maintained according to the changes in the environment, if rigid bodies move, by locally reconstructing the decomposition: in this case, first, the moved rigid bodies are removed from the lists of cluttering rigid bodies associated with the cells they do not intersect with (clutter) anymore. The list of rigid bodies associated with the newly cluttered cells are updated and the octree is then updated by merging and splitting cells if needed.

The process proposed has been designed to identify places according to their complexity for the path planning process. The modular architecture allows defining new place identification to complement the topological model.

Our place identification process has been inspired by region growing algorithms of computer vision [31]. It performs a breadth-first search of the free cells of the octree (Fig. 12) in the environment model. At each level of the octree, free cells connected to cells of upper levels forming an existing (and currently growing) place are aggregated into that place. The remaining free cells of the considered level of the octree are used to initialize new places; connected cells initialize a single place. Cells where two neighboring places overlap are used to constitute a border connecting those places. Fig. 12 illustrates the place identification process for three successive levels of the quadtree modelling geometrically the free space for our illustration case environment.

**The place semantics** The place identification process allows determining the semantic information concerning the complexity of the places and their cluttering. Indeed, the size of the cells used to create a given place defines its complexity. The earlier the place has been created, the lower its complexity is. For instance, in the example provided in Fig. 12, the blue places have a complexity level of 1, the turquoise ones a complexity level of 2 and the violet places have a complexity level of 3. In addition, if octree cells of a place are cluttered with mobile rigid bodies, the place is set as cluttered. However, to use the human expertise, at the end of the place identification process, a human

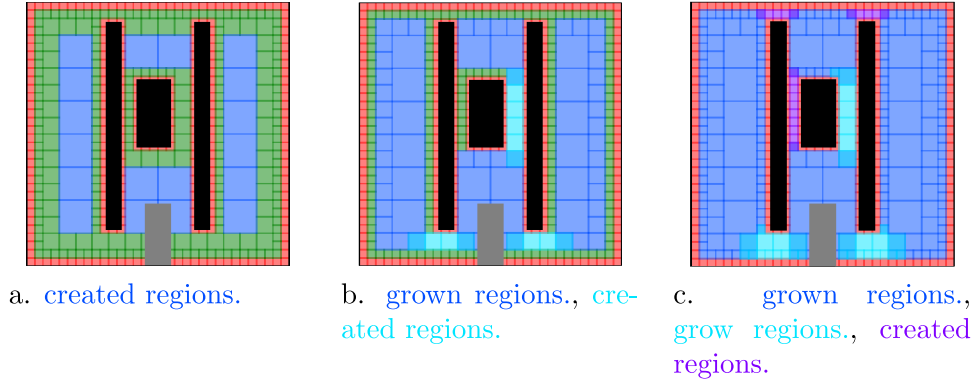


Fig. 12. Place identification process.

operator can manually edit the semantic information of the places.

### 5.2. Path planner

The modularity of our proposed path planning architecture is obtained by the definition of specific responsibility for the classes.

The semantic planner is in charge of the coordination of the planners of the other layers. In order to do so, the semantic planner uses a semantic interpreter which is able to perform logical reasoning based on the semantic information of the environment.

In Section 4.2, we explained that when solving the path planning query associated to a topological step, our path planning architecture selects path planning algorithms adapted to the local semantic context: the geometrical and local planners are chosen depending on the semantic information attached to the topological step involved. So far, and for the experiments described in Section 6, we have implemented an A\* algorithm for the geometrical planner, and an RRT algorithm for the local planner. Depending on the local semantic context (e.g. complexity to cross the place involved), a resolution of the path planning query by the geometrical planner may be helpful, or the use of the local planner in the 6D space (position and orientation) may be needed.

## 6. Experimental results

### 6.1. Compared strategies

The proposed strategy involving Geometrical, Topological and Semantic information (called GTS strategy hereafter) has been evaluated by comparing it to strategies involving only Geometrical and Topological information (GT strategy) or Geometrical information only (G strategy).

**G strategy** performs an RRT algorithm on the whole workspace. The number of random configurations is limited to  $10^6$  samples in order to limit the processing times.

**GT strategy** is the two-phase planning process described in Section 4 but the semantic information is not used. The costs of the topological graph are simply the distances between the borders  $C_{e_k} = d_k$ . Once the topological path has been computed, a geometrical milestone is associated to each border by using a RRT with a maximum number of 2000 random configurations. Then, the geometrical layer performs an RRT algorithm on each stepped of the topological path with a maximum number of  $10^5$  random configurations.

**GTS strategy** It is the complete strategy presented in Section 4. The semantically based costs of the topological graph are used during the coarse planning to favor the easiest places to cross. Moreover, the RRT algorithm is used when appropriate in cluttered places only; trajectories within uncluttered places are defined as linear

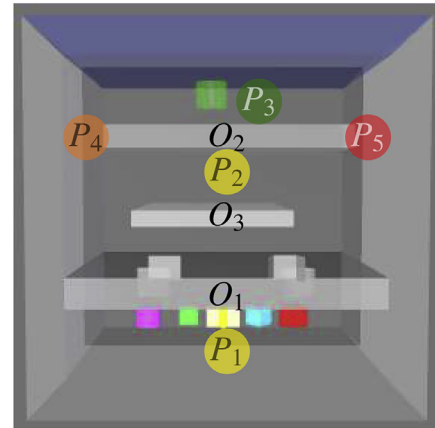
interpolations between the start and goal configurations of the corresponding topological step.

### 6.2. Case study

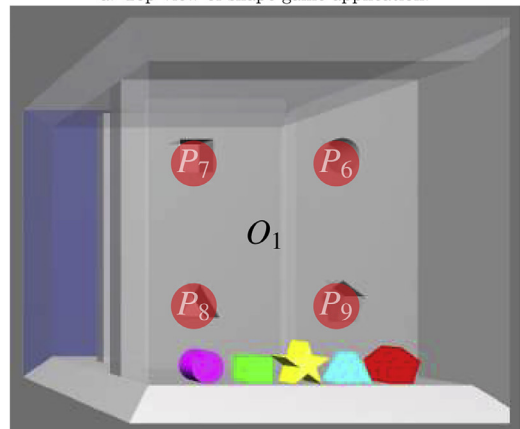
We have used the same 3D application to compare strategies G, GT and GTS. This application case is inspired by a “shape games” for babies and is presented in Fig. 13.

This case study is highly relevant to test, tune and validate our multi-layer environment modeling and path planning strategy; indeed:

- It is a very challenging application for path planning as it features the manipulation of objects in an highly geometrically constrained environment where traditional purely geometric path planners are



a. Top view of shape game application.



b. Front view of shape game application.

Fig. 13. 3D environment for experiments.

likely to show low efficiency;

- It is relevant to validate path planning strategies intended for the simulation of industrial manipulations, as (1) it deals with manipulating manufactured objects, having standard shapes which are representative of manufactured objects or parts to be assembled, and (2) moving those manufactured objects from the start to the goal configurations shown in Fig. 13 means moving them through very constrained passages or inserting them in properly shaped holes, which is representative of generic tasks to be performed when (e.g.) assembling, disassembling or maintaining an industrial system;
- Yet, the case study is not specific of any particular industrial task / application, therefore generic enough to validate our approach objectively.

The environment is made of a cubic workspace cluttered by 3 rigid bodies.  $O_1$  and  $O_2$  are static rigid bodies and split the workspace into 3 main places ( $P_1$ ,  $P_2$  and  $P_3$ ). The rigid body  $O_3$  clutters the place  $P_2$ . Places  $P_4$  and  $P_5$  are passages connecting  $P_2$  and  $P_3$  with different narrowness on the two sides of  $O_2$ . Several holes allowing passage from  $P_1$  to  $P_2$  through  $O_1$  are available ( $P_6$ ,  $P_7$ ,  $P_8$  and  $P_9$ ). These holes through  $O_1$  have different shapes.

The planning query considered here aims at bringing one of the colored shaped rigid bodies from  $P_1$  (Fig. 13b) to the green position in place  $P_3$  (Fig. 13a). While, for babies, playing this game commonly means finding the shape similarity between holes and objects, our planner has only to define a feasible trajectory.

When the GTS planning strategy is involved, the available semantic information describes information such as the *complexity* and the *cluttering* of the places, and the *mobility* of the rigid bodies (see Section 3), but also the *shapes* of both the places and the rigid bodies.

Two distinct cases of semantic information have been tested with the GTS strategy. These two cases describe the holes/rigid bodies *compatibility* with a boolean value specifying if a given rigid body can go through a given hole. In the first case (GTS1), only holes and rigid bodies having the same shape are compatible while in the second case (GTS2), the compatibilities between holes and rigid bodies are the ones described in Table 1

### 6.3. Experiments

The performance criteria for the evaluation of the G, GT, GTS1 and GTS2 strategies are the *success rates* and the *mean number of random configurations* used. The success rate is defined as the ratio between the number of successful trajectories from S to G and the number of tries. For the G strategy, the number of random configurations is the number of random configurations of the RRT algorithm. For the GT and GTS strategies, it is the sum of the number of configurations used during the coarse planning and of the ones used during the fine planning.

The strategies have been evaluated by running each planning strategy (G, GT, GTS1 and GTS2) for each manipulated object over one hundred times. The results of this evaluation are given in Fig. 14. In this figure, the bars represent the mean numbers of random configurations displayed on a logarithmic scale and the rates written above the bars

**Table 1**  
Holes/rigid bodies compatibility.

|   | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|-------|-------|-------|-------|
| ○ | ✓     | ✓     |       | ✓     |
| □ | ✓     | ✓     |       | ✓     |
| △ |       | ✓     | ✓     | ✓     |
| ⬠ |       |       |       | ✓     |
| ☆ |       |       |       | ✓     |

correspond to the success rates.

First of all, the GTS approach has to be compared with the usual RRT approach or G strategy :

- The GTS2 strategy always finds a trajectory whereas the G strategy fails for the Pentagon and Star objects.
- The mean total number of random configurations is always lower for the GTS strategies than for the G strategy, sometimes significantly (square, and triangle objects).

Concerning the GT strategy, it is worth noting that it may fail even when the G strategy succeeds. This is related to an important feature of the two-phase planning. Indeed, the topological path is a connected component of the free space obtained by the coarse planning phase. Depending on the manipulated object, the fine planning phase may fail (when there is no admissible path in the place between two milestones). In the GT strategy, the topological path is obtained by computing the shortest path between the borders. This may lead having to cross very narrow passages during the fine planning phase. Thus, it is necessary to add semantics to connexity and distance information.

This is what we observe in the results obtained using the GTS strategies. First, coarse planning based on semantics provides fine planning with a topological path that gives rise to a good success rate. Second, the more information available there is, the better the results are. Indeed, the GTS2 strategy defines feasible trajectories for all the objects while the GTS1 strategy does not succeed for the star object. It appears that controlling the path planning process with semantic data helps to choose relevant places to cross and thus eases the definition of geometrical milestones within the borders and the fine planning phase.

Therefore, as a conclusion on the global level, the GTS strategy leads clearly to better performances than the GT and G strategies: lower number of random configurations, lower processing times and better relevance of the computed paths, which validates our multi-layer approach.

However, the richness of these experiments calls for a deeper understanding of the processes involved. On one side, since probabilistic approaches have no proof of completeness in finite time, the results are really dependent on the maximum number of random configurations used in the algorithms. On the other side, splitting the path planning process into two phases requires a high control on the coarse planning to provide the fine planning with a useful topological path. Such a control is possible only with accurate semantic information. Next section will give detailed analysis and explain the behavioral artifact that may appear leading to unexpected low performances sometimes.

### 6.4. Detailed results analysis

Let us consider first the path planning results for the **square object** (Fig. 15).

First of all, for this object, the G strategy has a 100% success rate. This high success rate is mainly due to the compatibility this object has with several holes in obstacle  $O_1$  (cf. Table 1). Actually, most of the trajectories planned by the G strategy cross obstacle  $O_1$  through the pentagon hole (Fig. 15). However, the mean number of random configurations necessary to find such trajectory (almost  $1.4 \cdot 10^5$ ) points out the difficulties still encountered by the probabilistic planning techniques in finding feasible trajectories in such narrow passages.

Then, when introducing topological information with the GT strategy, one can notice that the path planner does not find any trajectory (0% success rate). This result is due to the introduction of the topological path definition performed during the coarse planning. This process aims at reducing the areas randomly explored, but in GT strategy, it is not accurately controlled. With this strategy, the only information available to select one path among all the holes of  $O_1$  is the distance. The path planner computes the shortest path which depends notably on the initial position of the colored object to be moved. For the

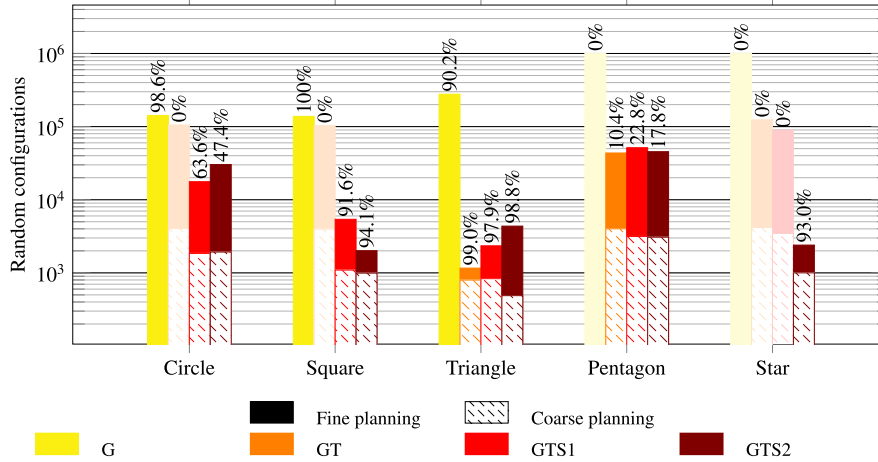


Fig. 14. Number of random configurations used and success rates for each planning query and planning strategy.

square object, the shortest path crosses obstacle  $O_1$  through the triangle hole which is not compatible with the square (Fig. 15b). Thus, the random exploration of the corresponding place necessarily fails. The lower number of random configurations used by the GT strategy compared to the G strategy is due to the  $10^5$  random configuration limit described earlier.

The GTS1 strategy introduces semantic information leading to a better control of the planning process and particularly the deterministic process of its coarse step. Indeed, the path planner has only the information that among all the holes of the obstacle  $O_1$ , the square object can go through the square hole, thus it selects this one (Fig. 15c). Moreover, the selected topological path passes by obstacle  $O_2$  through the largest passage. This control of the topological path selection allows defining the geometrical trajectory by using probabilistic processes with only  $5.4 \times 10^3$  random configurations (25 times less than the G strategy). The 91.6% success rate of this strategy means that the limitation of the number of random configurations chosen for the probabilistic processes is still a bit too low to solve the problem on time.

The GTS2 strategy involves the same path planning process as the GTS1 strategy but not the same semantic information. The GTS2 strategy uses a semantic information specifying a wider compatibility between objects and holes of  $O_1$ . Assuming that the hole/object compatibilities are expressed as booleans, the criterion allowing to select one hole among several compatible ones with one object is the distance (see (1)). For the square object, the selected topological path crosses obstacle  $O_1$  through the pentagon hole which is larger than the square one (Fig. 15d). This easier passage explains both the higher success rate of the GTS2 strategy comparing to the GTS1 strategy and the lower number of necessary random configurations.

Concerning the path planning results given in Fig. 14 for the different objects, we observe the following:

**Circle object:** The G strategy has an interesting success rate (98.6%) but needs more than  $10^5$  random configurations. It mainly finds a path crossing  $O_1$  through the pentagon hole. The GT strategy fails because it selects the topological path using the triangular hole (which is not compatible). The GTS1 strategy selects the round hole leading to a 63.6% success rate using less than  $2 \times 10^4$  random configurations while the GTS2 strategy selects the square hole and thus has a lower success rate (47.4%) and uses more random configurations ( $3 \times 10^4$ ).

**Triangle object:** The G strategy has a 90.2% success rate using more than  $2 \times 10^5$  random configurations. The GT strategy selects the pentagon hole for obstacle  $O_1$  and the narrow passage for obstacle  $O_2$  leading to a 99% success rate with about 1000 random configurations. The GTS1 strategy does not select the larger passage for

obstacle  $O_2$  but the triangular hole for obstacle  $O_1$ , leading to an increase in the random configurations needed (about 2000 mainly due to an increase of the random configuration used during the fine planning step) and a decrease of the success rate (97.9%). Finally, the GTS2 strategy selects the pentagon hole and the largest passage leading to an increase in the number of random configurations used (3000) but a decrease of those used during the coarse planning phase (divided by 2 if compared to the GTS1 strategy case). This increase of the number of random configurations is counterbalanced by an increase of the success rate (98.8%).

**Pentagon object:** The G strategy fails at finding a trajectory because all the holes through  $O_1$  are too narrow to be randomly found. The other strategies succeed and involve from 3 to  $4 \times 10^4$  random configurations. The GT strategy selects the pentagon hole and the narrow passage leading to a success rate of 10.4% while the GTS1 and GTS2 strategies reach a 20% success rate by selecting the pentagon hole and the large passage.

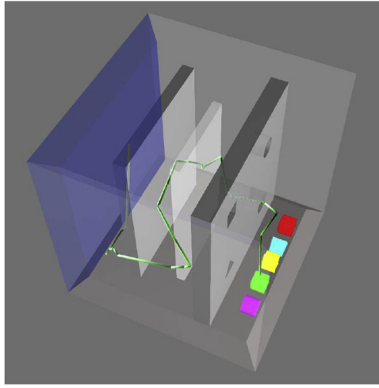
**Star object:** Both the G, GT and GTS1 strategies fail finding a trajectory for the star object. Indeed, the G strategy suffers the same weakness as for the pentagon object while strategies GT and GTS1 select the triangular hole (not compatible). In the GTS2 strategy the compatibility between the pentagon hole and the star object is set. Thus, it selects the topological path crossing  $O_1$  through the pentagon hole and using the largest passage to pass  $O_2$  leading to a 93% success rate while using only 2000 random configurations.

The performance of the path planner strongly depends on the topological path selected during the coarse planning phase. This coarse planning phase needs to be accurately controlled to get a relevant topological path. The modularity of the architecture allows proposing different strategies on this layer.

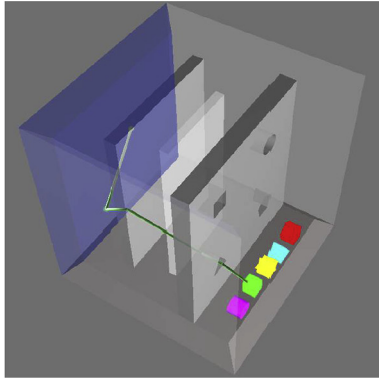
## 6.5. Application to an industrial case

After thoroughly validating our multi-layer (GTS) approach on the “shape game” study case, we applied it to a real industrial case presented in Fig. 16. The task considered is a maintenance one. It deals with the mounting (or unmounting) of a pump on (or from) a real power electronics subsystem. We first generated the environment model.

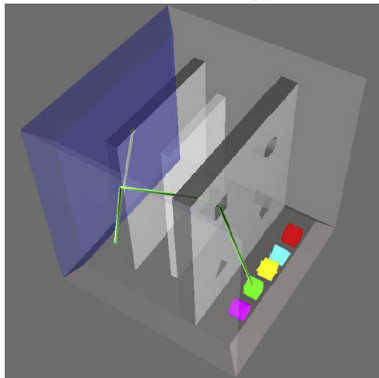
- The rigid bodies geometrical representations were directly derived from the CAD models of the pump and the power electronics subsystem considered, and the semantic level relates to the mobility (or not) of these objects.
- The free space geometrical and topological levels were built as



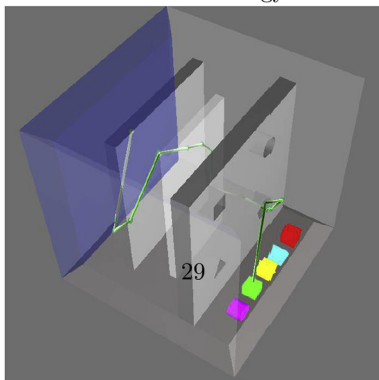
a. G strategy



b. GT strategy



c. GTS1 strategy



d. GTS2 strategy

Fig. 15. Planned paths for the square object.

explained in Sections 3.2 and 5.1.2. The semantic level describes the complexity to be crossed and the cluttering of the places, as defined in Section 5.1.2.

Then, we applied our multi-layer path planner and showed its ability to solve mounting and unmounting path planning requests using appropriately defined start and goal configurations in each case.

## 7. Conclusion and perspectives

In this paper, a novel approach for both environment modeling and path planning has been proposed. The contribution is two-fold:

1. Our novel multi-layer environment model involves geometrical, topological and semantic information in dedicated layers;
2. Our original path planning process takes advantage of our multi-layer environment model by controlling the two phases (exploration of the topology and geometrical trajectory computation) using semantics.

Introducing higher abstraction level information in the path planning process improves its performances with respect to state-of-the-art techniques:

- **From the qualitative point of view**, the relevance of the path obtained is improved. First, controlling the exploration of the topology using semantic information allows to rapidly find a relevant topological path. Second, during the fine planning phase, the path planning strategy is defined according to the local semantic context.
- **From the quantitative point of view**, the number of random configurations needed when using a probabilistic path planning method has been strongly reduced, mainly due to:
  - the semantic control of the coarse planning phase,
  - the fact that path planning queries are called at a local level (related to single topological steps of the coarse topological path),
  - the possibility, during the fine planning phase, to reduce even more the free space exploration by calling the geometric planner prior to the local planner.

However, our contribution is also open to improvements.

Concerning path planning techniques, enriching the set of planners involved in the fine planning phase will allow adapting even better the path planning strategy involved to the local semantic context.

The performances of the proposed path planning architecture depend strongly on the semantic information available. As an example, using scores instead of binary information for the object/hole compatibility would probably lead to better results than those presented in this paper.

Concerning the two-phases process, the deterministic coarse planning performed in the first phase needs to be finely controlled in order to avoid infeasible paths in the fine planning phase. As the whole path planning process may fail because of the coarse planning phase, it is necessary to improve the control upon the topological path definition process. On one side, in the current approach, both phases are performed once ; but it is also possible to put fine planning and coarse planning into an iterative process in order to be able to select another topological path when fine planning fails. On the other side, the modularity of our architecture will allow us to involve tools from cognitive sciences (ontologies, conceptual graphs,... [32,35,36]) to improve the richness and evolutivity of the semantic layer, and the way we acquire semantics, thus easing the capability of the multi-layer architecture to handle new application cases.

Moreover, the GTS architecture may also take advantage of the contribution of a human in the loop during the path planning process. Such an *interactive* path planner would directly benefit from the human expertise and from his perception of the task to perform at a high level of abstraction [9].

Finally, the proposition detailed in this paper has to be set in a larger perspective. This paper focuses on path planning but the proposed architecture is modular and evolute and aims at being

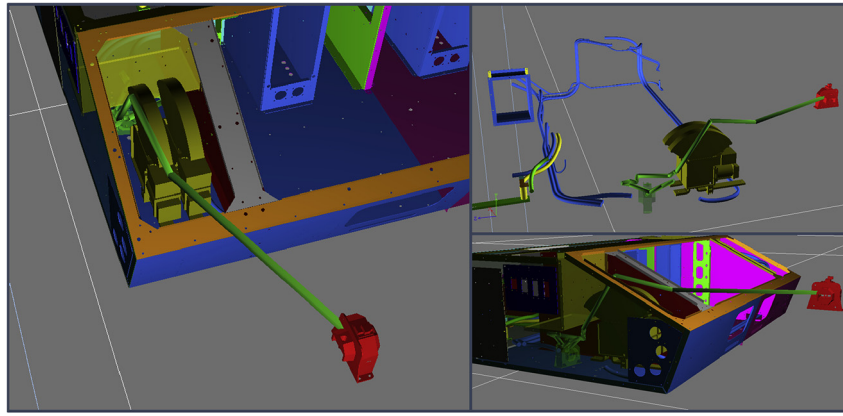


Fig. 16. Industrial case application.

associated to a wider range of functionalities. As an example, the proposed architecture could cooperate with task-planning tools such as those described in [33,34] in a combined task and motion planning approach [37,38]. The multi-layer architecture aims at introducing information and reasoning on high abstraction level information in all the processes involved.

## References

- [1] T. Lozano-Perez, Spatial planning: a configuration space approach, *Trans. Comput.* 100 (2) (1980) 108–120.
- [2] Jean-claude Latombe, *Robot motion planning*, Springer, 1991.
- [3] M.C. Howie, *Principles of robot motion: theory, algorithms, and implementations*, MIT press, 2005.
- [4] K. Oussama, Real-time obstacle avoidance for manipulation and mobile robots. *International Conference on Robotics and Automation*, volume 2, IEEE, 1985, pp. 500–505.
- [5] H.O. Mark, A random approach to motion planning, Utrecht University, Department of Computer Science, 1992 Technical report.
- [6] M.L. Steven, Rapidly-exploring random trees: a new tool for path planning. technical report, (1998). Iowa State University
- [7] M.A. Ahmadi-Pajouh, F. Towhidkhal, S. Gharibzadeh, M. Mashhadimalek, Path planning in the hippocampo-prefrontal cortex pathway: an adaptive model based receding horizon planner. *Med. Hypotheses* 68 (2007) 1411–1415.
- [8] M. Taix, D. Flavigné, E. Ferré, Human interaction with motion planning algorithm. *J. Intell. Rob. Syst.* 67 (3) (2012) 285–306. Springer
- [9] Y. Yan, E. Poisson, F. Bennis, An interactive motion planning framework that can learn from experience. 59 (2015) 23–38. Elsevier
- [10] Y. Morales, A. Watanabe, F. Ferreri, J. Even, T. Ikeda, K. Shinozawa, T. Miyashita, N. Hagita, Including human factors for planning comfortable paths. In *IEEE International Conference on Robotics and Automation (ICRA)*, (2015), pp. 6153–6159.
- [11] X. He, Y. Chen, Haptic-aided robot path planning based on virtual tele-operation. 25 (2009) 792–803.
- [12] N. Ladeveze, J.-Y. Fourquet, B. Puel, Interactive path planning for haptic assistance in assembly tasks. 34 (2010) 17–25. Elsevier
- [13] E. Bastianelli, D. Bloisi, R. Capobianco, G. Gemignani, L. Iocchi, D. Nardi, Knowledge representation for robots through human-robot interaction, 2013. ArXiv preprint arXiv:1307.7351.
- [14] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal, J. González, Multi-hierarchical semantic maps for mobile robotics. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2005), pp. 2278–2283.
- [15] O.M. Mozas, P. Jensfelt, H. Zender, G.-J.M. Kruijff, W. Burgard, An integrated system for conceptual spatial representations of indoor environments for mobile robots. *Proceedings of the IROS 2007 Workshop: From Sensors to Human Spatial Concepts (FS2HSC)*, (2007), pp. 25–32.
- [16] A. Pronobis, O.M. Mozas, B. Caputo, P. Jensfelt, Multi-modal semantic place classification. 29 (2010) 298–320.
- [17] P. Viswanathan, T. Southey, J. Little, A. Mackworth, Place classification using visual object categorization and global information. In *IEEE Canadian Conference on Computer and Robot Vision*, (2011), pp. 1–7.
- [18] J. Vogel, B. Schiele, A semantic typicality measure for natural scene categorization. *Pattern Recognition Lecture Notes in Computer Science*, volume 3175, Springer, 2004, pp. 195–203.
- [19] A. Torralba, K.P. Murphy, W.T. Freeman, A.R. Mark, Context-based vision system for place and object recognition. *IEEE International Conference on Computer Vision*, (2003), pp. 273–280.
- [20] I. Ulrich, I. Nourbakhsh, Appearance-based place recognition for topological localization. *IEEE International Conference on Robotics and Automation*, volume 2, (2000), pp. 1023–1029.
- [21] M. Devy, R. Chatila, P. Fillatreau, S. Lacroix, F. Nashashibi, On autonomous navigation in a natural environment. 16 (1995) 5–16.
- [22] C. Galindo, J.-A. Fernández-Madrigal, J. González, A. Saffiotti, Using semantic information for improving efficiency of robot task planning. *ICRA Workshop: Semantic Information in Robotics*, (2007). Rome, Italy
- [23] C. Galindo, J.-A. Fernández-Madrigal, J. González, A. Saffiotti, Robot task planning using semantic maps. 56 (2008) 955–966.
- [24] R. Iacob, D. Popescu, P. Mitrouchev, Assembly/disassembly analysis and modeling techniques: a review. 58 (2012) 653–664.
- [25] S. Jeong, D. Wegner, S. Noh, Validation of an ontology-based approach for enhancing human simulation in general assembly environments. *Proceedings of the WCE*, volume 3, (2010).
- [26] C. Dornhege, M. Gissler, M. Teschner, B. Nebel, Integrating symbolic and geometric planning for mobile manipulation. In *IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR 2009)*, (2009), pp. 1–6.
- [27] R. Philippsen, N. Nejati, L. Sents, Bridging the gap between semantic planning and continuous control for mobile manipulation using a graph-based world representation. 1st Int. Work. on Hybrid Control of Autonomous Systems held in conj. with the Int. Joint Conf. on AI, Pasadena, USA, (2009).
- [28] E. Tosello, Z. Fan, E. Pagello, 2016, A semantic knowledge base for cognitive robotics manipulation, University of Padova.
- [29] S. Hanan, The quadtree and related hierarchical data structures. 16 (1984) 187–260.
- [30] H.H. Chen, T.S. Huang, A survey of construction and manipulation of octrees. 43 (1988) 409–431.
- [31] R. Jain, R. Kasturi, G.S. Brian, *Machine Vision*, McGraw Hill, 1995.
- [32] F.V. Harmelen, V. Lifschitz, B. Porter, *Handbook of knowledge representation*, Elsevier, 2008.
- [33] B. Bonet, H. Geffner, *Planning as heuristic search: new results*. European Conference on Planning (1999) 360–372. Springer
- [34] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*. Artificial Intelligence, volume 25, Prentice-Hall, Englewood Cliffs, 1995.
- [35] R. Arp, B. Smith, A.D. Spear, *Building ontologies with basic formal ontology*, MIT Press, 2015.
- [36] A.Y. Ng, S.J. Russell, Algorithms for inverse reinforcement learning. *Proceedings of ICML* (2000) 663–670.
- [37] L.P. Kaelbling, T. Lozano-Pérez, Hierarchical task and motion planning in the now. In *IEEE International Conference on Robotics and Automation (ICRA)*, (2011), pp. 1470–1477.
- [38] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, P. Abbeel, Combined task and motion planning through an extensible planner-independent interface layer. In *IEEE International Conference on Robotics and Automation (ICRA)*, (2014), pp. 639–646.