To link to this article: DOI:10.1007/s12008-015-0272-5
URL: http://dx.doi.org/10.1007/s12008-015-0272-5

To cite this version:
Cailhol, Simon and Fillatreau, Philippe and Zhao, Yingshen and Fourquet, Jean-Yves *A hierarchic approach for path planning in Virtual Reality*. (2015) International Journal on Interactive Design and Manufacturing (IJIDeM), vol. 9. ISSN 1955-2513

A hierarchic approach for path planning in Virtual Reality

Simon Cailhol   Philippe Fillatreau   Yingshen Zhao   Jean-Yves Fourquet

March 20, 2015

Abstract

This work considers path-planning processes for manipulation tasks such as assembly, maintenance or disassembly in a virtual reality (VR) context. The approach consists in providing a collaborative system associating a user immersed in VR and an automatic path planning process. It is based on semantic, topological and geometric representations of the environment and the planning process is split in two phases: coarse and fine planning. The automatic planner suggests a path to the user and guides him trough a haptic device. The user can escape from the proposed solution if he wants to explore a possible better way. In this case, the interactive system detects the users intention and computes in real-time a new path starting from the users guess. Experiments illustrate the different aspects of the approach: multi-representation of the environment, path planning process, users intent prediction and control sharing.

1 Introduction

The industrial product development process is going faster and faster with more and more complex products. This leads to a need of tools allowing to rapidly test a product at all the PLM stages during the design phase. Performing such tests with virtual prototypes accelerates the design process while reducing its cost. There is a particular need for performing the tasks that involve human operator manipulation with virtual prototypes. Here comes the interest of Virtual Reality (VR) to run these tests [11].

Among these tasks, VR community has studied assembly/disassembly, dismantling and maintenance. To perform such tasks in VR simulation, the means for interaction allowing manipulating the CAD models is one of the main issues [12, 5, 18]. Another issue is to identify the system mechanical constraints to provide the VR operator with an assembly/disassembly plan [14, 15].

The main issue of such tasks is to find paths for the systems CAD models. For this issue of such applications, we propose a collaborative path-finding system based on the interaction of a user immersed in a VR simulation and an automatic path planning process inspired from robotics.

Collaboration is defined as follows. The system provides a initial planned path and the user is guided along a computed trajectory through an haptic device. However, the user can disagree the proposed path and try to go in another direction. The system must compute a new path every time the user tries to test another solution. Thus, it must be able to take into account the users interactions in real time to update the suggested path and it requires control sharing between the user and the planner while performing the task.

Robotics path planners mainly deal with geometric aspects of the environment. The VR context of our planner involves a human in the loop with a different environment representation. Thus, we chose to split the planning process in two phases: a coarse planning dealing with topological and semantic models of the environment (the places, their semantics and their connectivity) and a fine planning dealing with geometry and semantics (geometry of obstacles and places and their complexity). This planning process partitioning provides a framework compatible with the human path planning process described in [3]. Thus, the proposed interactive path planner is based on a multi-layer environment representation (semantic, topological and geometric). All these environment models are used by distinct planner layers to perform the coarse (semantic and topological aspects) and fine (semantic and geometric aspects) planning and to assist VR user.

Table 1: Main path planning methods

	Global approaches	Local approaches
Deterministic strategies	Cells decomposition, Roadmap	Potential field
Probabilistic strategies	Probabilistic RoadMap	Rapidly-exploring Random Tree, Rapidly-exploring Dense Tree

2 State of the art

2.1 Automatic path planning

The automatic path planning issue has been deeply studied in robotics. These works are strongly based on the Configuration Space (CS) model proposed by [20]. This model aims at describing the environment from a robots Degrees of Freedom (DoF) point of view. The robot is described using a vector where each dimension represents one of his DoF. A value of this vector is called a configuration. So, all the possible values of this vector form the CS. This CS can be split into free space and colliding space (where the robot collides with obstacles of the environment). With this model, the path planning from a start point to a goal point consists in finding a trajectory in the free space between these two points in the CS.

The main methods for path planning are given in the Table 1 where we distinguished the deterministic from the probabilistic ones, but also, the ones involving global approach from the ones involving local one. More details on path planning algorithms and techniques are available in [7, 17].

2.2 Control Sharing

There already exist some applications involving path planners with human interactions (robot tele-operation, semi-autonomous vehicles, virtual environment exploration,...). These applications allow us to identify two aspects in control sharing:

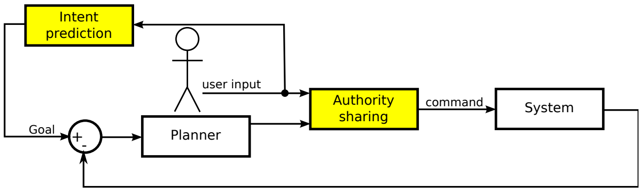


Figure 1: Sharing control model in semi-automated systems (yellow boxes are responsible for the control sharing)

- Authority sharing: it aims at defining how the authority on the system is shared between automatic planner and human. To deal with this issue, different strategies can be found in the literature. The use of virtual fixtures [21], authority switched to robot for fine motion operations [2], authority progressively transferred to robot while reaching the goal [23], for an anthropomorphic robot, Cartesian (position and orientation of end effector) control by user and joint control by planner [24]. The authority sharing through haptic devices were studied for semi-autonomous vehicles driving. In this case, from the horse riding experience, [13] suggests to use an haptic interface with a H-mode to perceive users involvement and allocate the authority according to it (the more the user is involved, the more authority he has).

- Intent prediction: it aims at predicting the intent of the human to define the goal of an automatic controller and so to assist the human performing the task. These techniques are strongly based on behavior or trajectory recognition [1, 10, 18, 25], on minimum jerk criterion [23], on model predictive control [4, 19]. Dragan also recently proposed to find the targeted goal among a set of potential ones from the current movement direction [9].

We summarize these two control sharing aspects in Fig 1 where the yellow boxes illustrate the control sharing. These techniques allow involving human and automatic planning system to perform a task. However, the users actions do not affect the automatic planner strategy to compute the path.

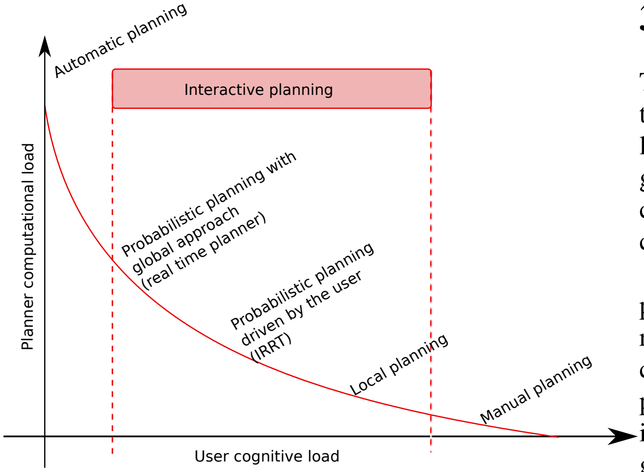


Figure 2: Computational vs cognitive load for interactive planning

2.3 Interactive path planning

Some works propose collaboration between a human operator and an automatic planner in the path planning process. The simpler one [16] uses a potential field strategy. An attractive field to the goal is computed and used to guide the user through a haptic device. Another interactive planner from [16] guides the user along a computed trajectory. To compute this trajectory in real time, a cell decomposition of the free space is used to define a 3D tunnel. Then a RDT algorithm computes a path within this 3D tunnel. The whole trajectory computation process is restarted if user goes away the proposed trajectory. Finally, an interactive planner build from a probabilistic strategy [22] uses the users action to constraint the random sampling of the configuration space in the RRT growing.

These three planners do not involve the human user in the same way. The first one gives a strong responsibility to the user (its up to him to deal with the obstacles and to avoid collisions). The second one suggests a whole trajectory the user can go away from to restart the whole planning process. The last one allows the user to point a direction that gives to the planner a preferred direction to explore. These differences of roles induce different computational load for the planner and cognitive load for the user as illustrated in Fig 2.

3 Proposed interactive planner

This section presents the concepts of the strategy used in the interactive planner shown in the Fig 3 where colors are linked to the environment and planning layers: yellow for geometry, orange for topology and red for semantics. The concepts used are illustrated here with 2D illustrations for clarity, but the model stands identical for 3D simulations.

We argue that involving semantic and topological aspects in path planning in addition to the common geometric ones allows adapting the planning strategy to the local complexity of the environment. To deal with it, a coarse planning is performed first using semantic and topological information. Then, heavy geometric path planning strategies are used merely locally, (according to the place complexity). This allows us to plan path in real time (without disturbing users immersion in the VR simulation), and to take into account users action while performing the task to interactively update the planned path.

3.1 Environment representation

The environment example given in Fig 4.a to illustrate the concepts is made of a square workspace cluttered by 3 fixed obstacles (O_1 to O_3) and 1 moving obstacle (O_4).

The semantic layer of environment representation given in Fig 3 is made of *Places*. A *Place* is identified (Fig 4.b) taking into account only the static obstacles to build a static representation of the environment. Semantic attributes are assigned to the *Places* to describe their complexity (size, shape, cluttering,...) for path planning.

The topological layer connects the *Place*s through *Border*s. A *Border* $B_{i,j}$ represents the overlapping area of *Place*s P_i and P_j. These *Border*s are then used to build the *Topological graph* of the environment shown in Fig 4.c. In this *Topological graph*, the nodes correspond to the *Border*s, and the edges to *Place*s (and so to their semantics). The Fig 5.a shows the distance between the *Border*s centers in *Place* P_4. These distances are set to the edges of the *Topological graph* as attributes (Fig 5.b for *Place* P_4). In Fig 4.c, the *Place* attribute corresponding to the edges are given in a color linked to their complexity (from (green) low to very high (red)).

The geometric environment representation consists in a geometric description of the obstacles surfaces using meshes (*Obstacle* and *Environment obstacle*). Then, a cell

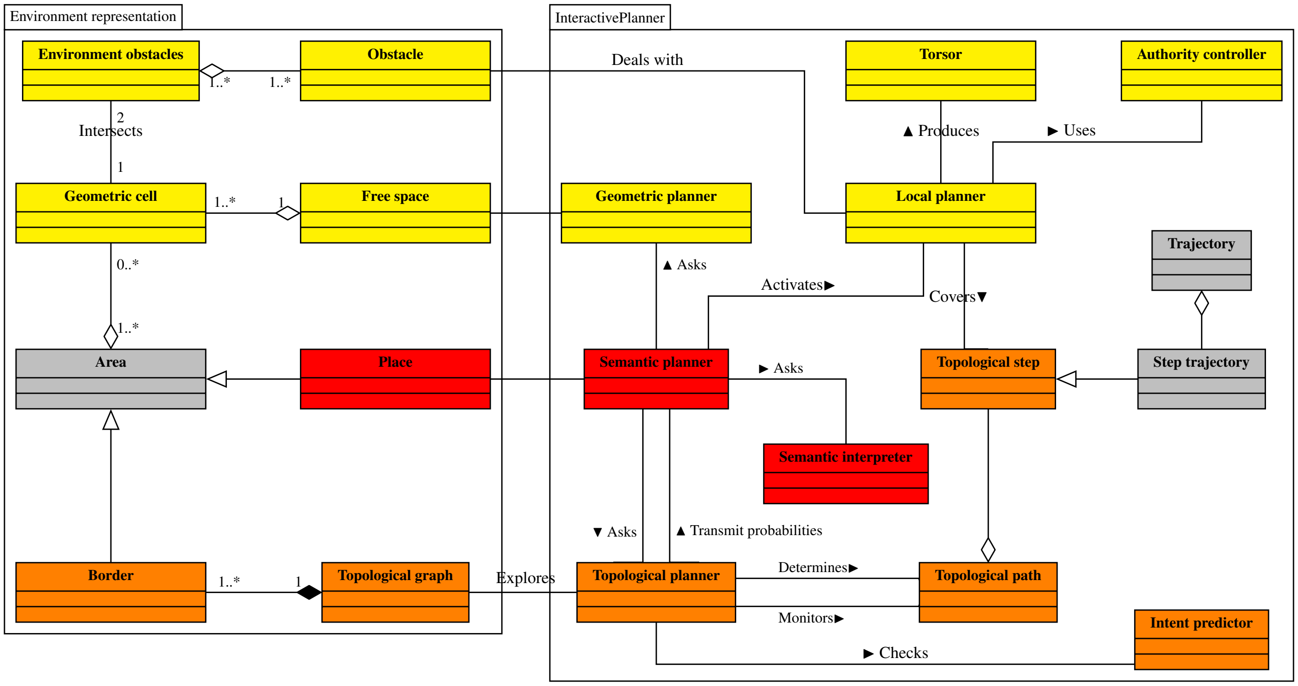
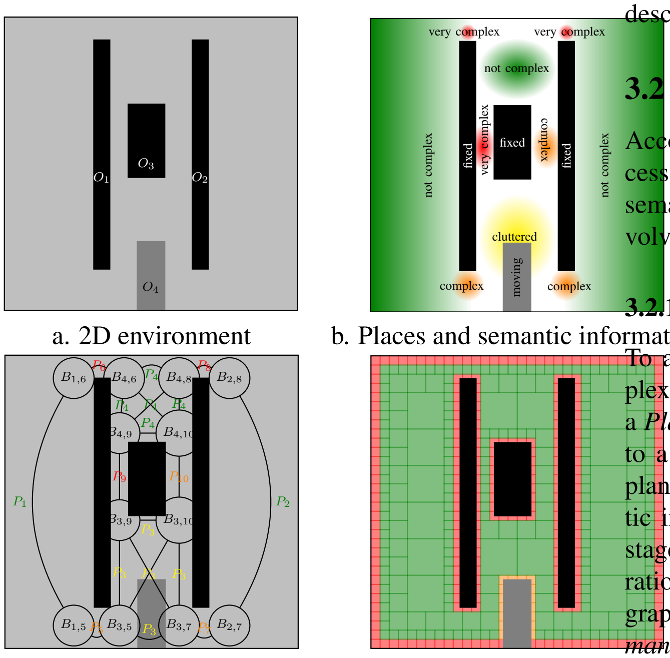


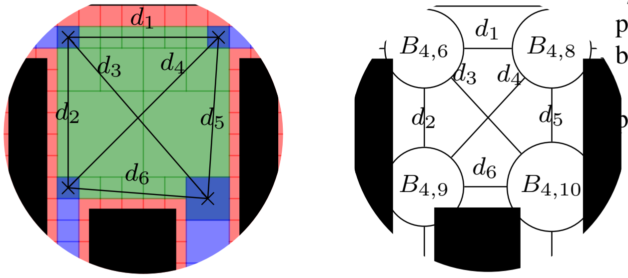
Figure 3: UML domain model of environment representation and interactive planner (red classes handle semantics, orange classes handle topology, and yellow classes handle geometry)



a. 2D environment b. Places and semantic information

c. Topological graph d. Free space decomposition

Figure 4: Different representations of the environment



a. Distances in place P_4 b.Topological graph of place P_4

Figure 5: Topological graph of place P_4

decomposition of the environment (*Free space* and *Geometric Cell*) is made with a quadtree (an octree in 3D) to describe the 2D (3D) free space (Fig 4.d).

3.2 Planning aspects

According to these environment models, the planning process is split in two stages: the coarse planning involving semantic and topological layers and the fine planning involving semantic and geometric layers.

3.2.1 Coarse planning

To adapt the geometric planning strategy to local complexity, the whole path is split in steps. A step refers to a *Place* of environment representation. A step also refers to a *Border* to reach to fulfill the step. The geometric planning strategy is thus chosen according to the semantic information of the steps place. To run this planning stage, two nodes corresponding to start and goal configurations are added to the *Topological graph*. To direct the graph exploration the *Semantic planner*, based on the *Semantic interpreter*, assigns costs (C) to graphs nodes ($n_{i,j}$) and edges (e_k) accordingly to the semantic information of involved places (1).

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

Where $sem(P)$ is the semantic information of *Place* P, e_k is a graphs edge, d_k its distance attribute and $P(ek)$ its place attribute; $n_{i,j}$ is the node linked to the border $B_{i,j}$ between P_i and P_j.

These costs make the cost of a path (C_{path}) computation possible (2).

$$C_{path} = \sum_{n_{i,j} \in path} C_{n_{i,j}} + \sum_{e_k \in path} C_{e_k} \tag{2}$$

Then the graph is explored by the *Topological planner* using a Dijkstra algorithm [8] to find the less expensive *Topological path* between start and goal nodes. This *Topological path* is used to split the trajectory in *Topological steps*, each step corresponding to a *Place* to cross (a edge of *Topological path*) and a *Border* to reach (a node of *Topological path*).

3.2.2 Fine planning

This planning stage consists in finding the concrete geometrical path. To do so, each *Topological step* is used to define a milestone configuration within the border to reach. Then, accordingly to the semantic information of the *Place* to cross, we adapt the geometric path planning strategy. Depending on the cluttering semantic attribute, an RRT algorithm can be ran on the place to set intermediate milestones within the step. When all the milestones have been defined, the *Local planner* guides the user toward the next milestone. This one compute a *Torsor* from a linear interpolation applied on a haptic device.

3.2.3 Coarse and fine planning organization

The coarse and fine planning are used to manage the whole planning. The *Topological path* and its steps are concepts allowing the different planning layers to share the information. Once the *Topological path* found and the *Topological step*s defined, the steps information is used by the *Semantic planner* to accurately set the geometric layer.

3.2.4 Process monitoring

While the user is performing the task, he is guided toward the next milestone configuration through the haptic device. This next milestone must be updated while the user is progressing along the path. On the geometric layer, the next milestone is set to the Local planner for the guidance computation when the current one is considered as reached. The goal is considered as reached when the distance between the goal and the current position is smaller than θ_d. On the topological layer, the milestone is a Border, so even if the user is guided toward a geometric configuration set within the *Border*, the milestone is considered as reached as soon as the user enters the *Border*. Thus, when the target *Border* is reached, the next *Topological step* is used to set the *Local planner* and to monitor the remaining process.

3.3 Control sharing aspects

The planner provides user with a guidance *Torsor* through the haptic device used for object manipulation. The *Local planner* computes this guidance *Torsor*. For each layer of such planner architecture, specific ways to share control can be envisaged as shown in Table 2.

Table 2: Interaction means on the different layers

	Authority sharing	Intent prediction
Semantic layer	Learn from users action new semantics information or means to deal with them to accurately set the topological and geometric planners	Interpret planning query expressed in natural language (*assemble this part on this one*, *bring this object on this one*,...)
Topological layer	Check if user agrees the proposed topological path. Trying to predict his intents on the topological layer (which place he is targeting)	Learn the kind of paces the user prefers to cross to advantage them during the topological path planning process
Geometric layer	Dynamically balance the authority on the object manipulation (between human and automatic planner) by modulating the automatic planner guidance force	Find the targeted next place to redefine the geometric planner goal

In Table 2 it appears that the intent prediction for the geometric layer is directly linked to the authority sharing of topological layer. Indeed, within a *Place*, the set of potential goals to get out of this *Place* is made of the corresponding *Border*s. The intent prediction is made with geometric movement and geometric information on *Border*s. The re-planning is made by the *Topological planner* for a new *Topological path* definition.

The same applies for the intent prediction of the topological layer and the authority sharing of the semantic layer. Cost functions of (1) may be learned from the *Place*s the user prefers to cross. These preferred *Place*s attributes can be identified from all the re-planning done due to users action. The new cost value defined with these functions will thus change all the incoming topological re-planning.

The control sharing of the proposed planning architecture is focused on the geometric and topological layers. A H-mode from [13] had been implemented for geometric authority control. An intent prediction inspired from [9] had also been set to make the topological path re-planning available.

4 Implementation

4.1 VR platform

We implemented this architecture in Virtools^{TM}4.1 software through libraries developed in C++ language. We developed 3 distinct libraries: 2 autonomous libraries corresponding to environment model and path planner and an interface library. The VR devices used are a large 3D display to immerse interacting user into the environment; a motion capture system to consider user's point of view for scene display and to allow him to visually explore the environment; and a haptic arm to manipulate the objects. Fig 6 shows the corresponding VR platform. The devices used in these simulations are:

- A large passive stereo-visualization screen
- A ART motion capture system ARTTRACK2
- A Haption virtuose 6D 35-45 haptic arm

4.2 Environment representation built

The environment model is implemented in a dedicated library interfaced to Virtools^{TM} with a specific library.

The environment representation we use is made of 4 models:

- The objects of the environment represented through meshes and positioning frames. We distinguish among them the fixed ones and the moving ones using a flag parameter to be able to exclude the moving ones while identifying the places (static mapping of the environment). All the objects of the environment can have semantic attributes assigned. This part is strongly based on the CGAL project [6].
- The free space description through an octree decomposition of the 3D scene (in this case also, the nodes colliding with fixed object are distinguished from those colliding with only moving objects)
- The topological graph to model the places connectivity (the graphs nodes are the borders, and the edges the places)
- The set of places and their borders. We defined some procedure to automatically identify the places from the octree structure and the semantic information attachment is manually made, choosing for each place the right attributes among a set of available ones. One attribute is automatically set: cluttered if the place contain moving obstacles

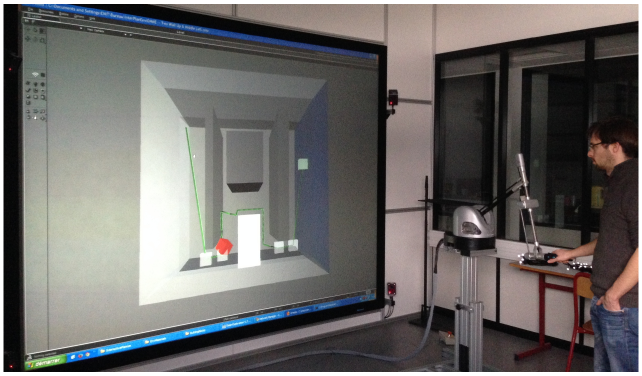


Figure 6: Simulation on VR platform

The attributes available in our simulations allow describing the level of complexity of a place: low, average, high, and very high.

4.3 Planner implementation

The planner is also implemented in a dedicated library and interfaced to Virtools^{TM} using the same interface library used to interface the environment.

4.3.1 Planning classes

Four classes have been defined corresponding to the four planners. Each of these planner classes deals with an environment model. The *Local Planner* provides the user with the guidance. The *Geometric planner* finds, if necessary, a geometrical path to cross a cluttered *Place*s. The *Topological planner* explores the *Topological graph* to build the path and the steps managed by the local and the geometric planners. The *Semantic planner* coordinates the whole planning process, asking the *Topological planner* for the *Topological path* and planning which strategy will be used on the geometric layer. For the weights computation, we defined the function of (1) assigning the weights as given in (3).

$$\begin{aligned} C_{n_{i,j}} &= \frac{C_{complexity}}{2} \\ C_{e_k} &= d_k.C_{complexity} \end{aligned} \tag{3}$$

Where $C_{complexity}$ is set according to the involved *Place*s' complexity to 0, 0.5, 1 and 5 for low, average, high and very high complexity.

4.3.2 Control sharing classes

Two main classes improve the planner for the control sharing. The first one is correspond to the *Authority controller*. It aims at modulating the guidance norm corresponding to user involvement. It allows user to feel free when he is exploring for others ways. The second one is the *Intent predictor*. It detects the user intents to compute a new *Topological path* when the user goes away the proposed one. These two classes and there computation are strongly based on the instantaneous movement computed from the recorded trajectory in *Trajectory* and *Step trajectory* objects.

5 Simulation and results

We have implemented the following simulations on our VR platform (Fig. 7). The VR devices used here are a large screen using passive stereoscopy for the 3D visualization and immersion, an AR Track system for the user view-point capture and a Virtuose 6D 35-45 haptic device for the part handling.

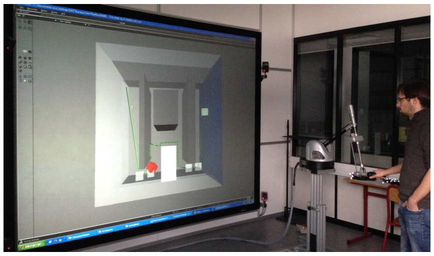


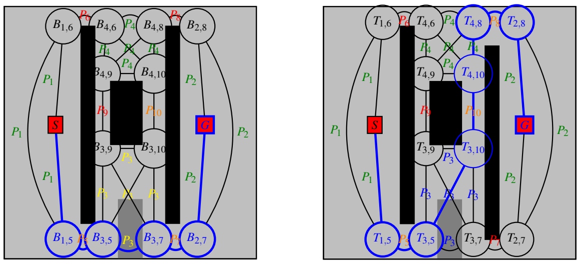
Figure 7: Simulation on VR platform

The first simulation is a 3D instance of the 2D example used to illustrate the principles of our planning strategy in section 3. It has been used for development and allowed to test the collaboration of the planners. The second simulation shows a richer semantics of the environment (semantic attributes that describe the shape of objects and places). This allows showing how the control of the planning process, using the semantic information, increases the reliability of the planned path while reducing the processing time.

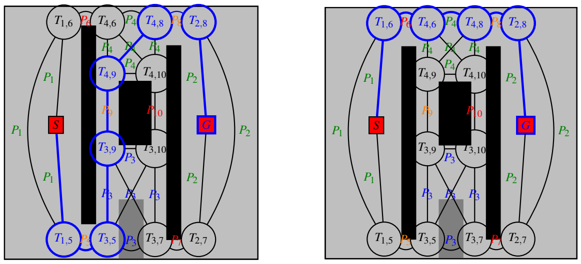
5.1 Semantic control and control sharing application

5.1.1 Simulation scene

To test the multi-layer structure on the laboratory's VR platform, the environment used is a 3D instance of the environment given in section 3. This environment is a cubic workspace with four obstacles cluttering the scene (3 fixed and 1 moving). Different environment configurations have been tested moving the fixed obstacles to change the complex passages locations (O_1 and O_2 are moved vertically and O_3 horizontally). The corresponding topological graphs are given in Fig. 8. This figure also illustrates the planning query in these environments. It aims at bringing a virtual object from a start point S in place P_1 to a goal point G in place P_2. The topological paths found by the topological planner are also displayed in bold blue lines in the topological graphs.

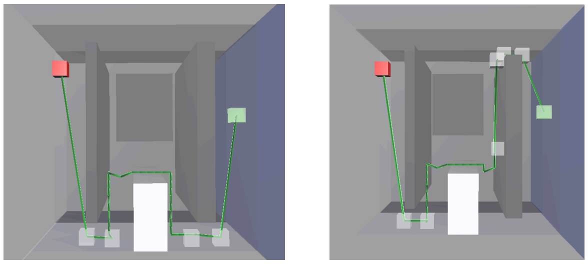


a.Environment 1 b.Environment 2

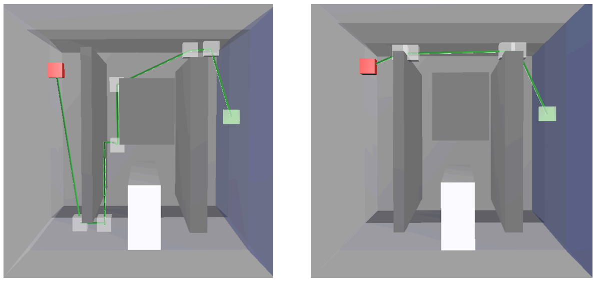


c.Environment 3 d.Environment 4

Figure 8: Experimental environments



a.Environment 1. b.Environment 2.



c.Environment 3. d.Environment 4.

Figure 9: Planning results

5.1.2 Path planning

Figure 9 shows the real path computed in the environment illustrated in Fig. 8. The manipulated object is the red cube and the targeted goal is the green one. The path is displayed in green. The paths on place P_3 avoid the mobile obstacle O_4 because of the RRT algorithm performed on this cluttered place. The number of random configuration used to find such paths is given in Fig.10. Depending on the crossed places and the paths defined by the RRT algorithm the number of random configuration needed is between 21 (Environment 4) and 129 (Environment 1). When using the RRT algorithm only, defining similar path needs from 1529 (Environment 4) to 8939 (Environment 3) random configurations.

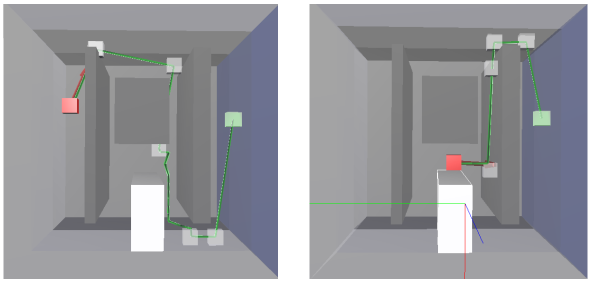
5.1.3 Path re-planning

Figure 11 illustrates the topological re-planning including real-time prediction of user's intent. In Fig. 11.a, in the first step, the user seems to prefer the narrow passage. Predicting it, the topological path is recomputed taking into account this intent. In Fig. 11.b, the user doesn't follow the guidance along the geometrical path in the third topological step. Thus, the topological planner computes a new topological path. The path re-planning including fine planning process is performed enough fast to allow real-time interaction.

5.2 Shape as semantic information

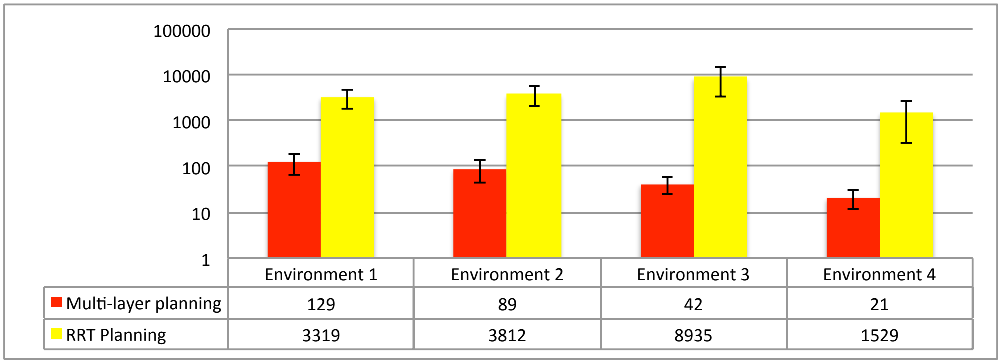
5.2.1 Simulation scene

The simulation scene (Fig. 12) is made of a cubic workspace divided in three large places by two walls (Fig. 12.a). The wall in the foreground is an obstacle with



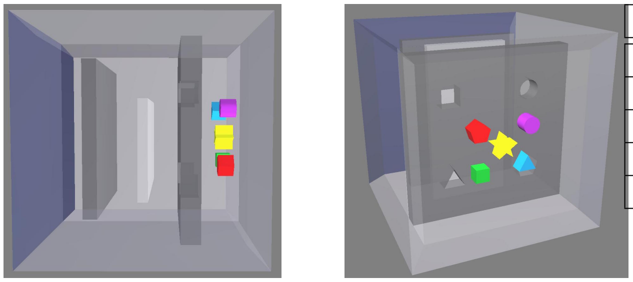
a.Re-planning in step 1. b.Re-planning in step 3.

Figure 11: Topological re-planning in environment 1



	Environment 1	Environment 2	Environment 3	Environment 4
🟥 Multi-layer planning	129	89	42	21
🟨 RRT Planning	3319	3812	8935	1529

Figure 10: Average and standard deviation of number of random configurations used for path planning in the design application



a.Scene top view.

b.Pierced wall.

Figure 12: Shape application environment

Object\ Hole	Circle	Square	Triangle	Pentagon
Circle	✓	✓		✓
Square	✓	✓		✓
Triangle		✓	✓	✓
Pentagon				✓
Star				✓

Table 3: Object/hole compatibility

four holes (Fig. 12.b). Each hole has a characteristic shape (square, triangular, round and pentagonal). The wall in the background is an obstacle leaving a passage on each side (a large one on the bottom and a narrow one on the top in Fig. 12.a). A moving obstacle clutters the place between these two walls.

The topological places of this environment are: the three large places, the two passages around the background wall (Fig. 12.a), and the holes through the foreground wall, each hole corresponds to a place (Fig. 12.b). The semantic attributes attached to the places are: ”low complexity” for the three large places; ”high complexity” for the large passage around the background wall, and ”very high” for the narrow one. The additional ”cluttered” semantic attribute is assigned to the places containing moving objects. Attributes are also set to the wall holes to describe their shape (”square”, ”triangular”, ”round” and ”pentagonal”). These shape attributes allow the automatic path planner finding an accurate topological path being guided according to the semantic information. However, to provide the VR user with topological path alternatives, shaped objects can cross the wall through differently shaped holes. The corresponding object/hole compatibility is given in Table 3.

The planning query here consists in passing the two walls to move one of the shaped object (colored) from one side of the cube to the other.

5.2.2 Path planning

Figure 13 shows the path (green rays) computed for the triangular object with our proposed architecture. Whatever the shape of the object (except the star object) is, the path planned crosses the first wall through the hole with the same shape. For the star object, as there is no hole with the same shape, the *Topological path* crosses the triangu-

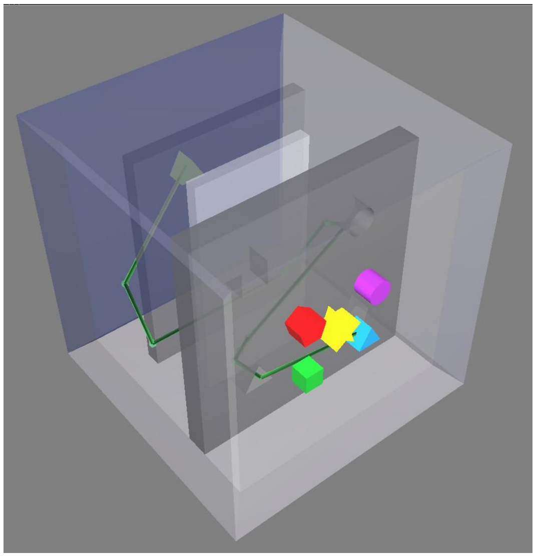


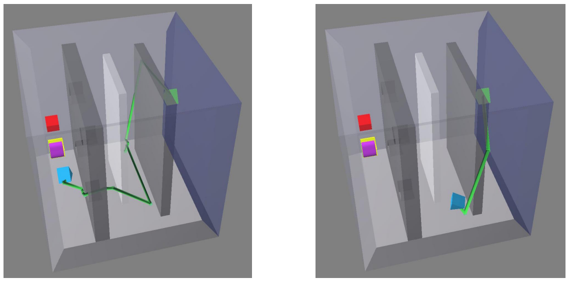
Figure 13: Planning results for triangular object

lar hole and thus provide an inaccurate path that doesn't allow to define non colliding path. Thus, to handle this query, the VR user have to move toward the pentagon hole to start a re-planning guided toward this shaped hole.

The number of configuration randomly defined to find a path for each objects is given in Fig.14. To get such results, for the multi-layer planning, the non collision constraint have been relaxed in the topological milestones (explaining that the automatic path planner find a path with colliding milestones in the triangular hole for the star object). For the RRT planning, the number of random configuration used have been limited to one million. Thus, for the RRT planning the nearer to one million the average number of random configurations is, the higher the failing rate is.

5.2.3 Path re-planning

Figure 15 illustrates the path re-planning in the case of the triangular object manipulation. Here, the user did not follow the haptic guidance along the geometrical path. In the first place, he moved toward the pentagonal hole starting thus the re-planning process. The resulting path goes through the pentagonal hole (Fig.15.a). In the middle place between the two walls, he moved toward the narrow passage, then the re-planned path goes that way (Fig.15.b). Thus, in both cases, a new multi-layer path planning is performed to take into account the operator's intents. Once the new *Topological path* defined the guidance is updated to assist the operator along is preferred path.



a.From first place.

b.From middle place.

Figure 15: Interactive path re-planning

6 Conclusion

This paper presents a novel multi-layer architecture for interactive path planning in VR simulations. This architecture is based on a multi-layer environment model and a multi-layer planner. Each layer deals with specific information (semantic, topological and geometric). The contribution of such an architecture is two-fold :

- First, it provides the user with real-time manipulation guidance by involving the semantic and topological information in the path planning process. The path planning process is accelerated by splitting the path in steps and then by adapting the geometric planning strategy to the local complexity of each step.
- Second, it integrates efficiently a human in the loop: path re-planning is computed based on real-time user's intent prediction and motion control is shared by the user and the planner.

The interest of such a planner architecture had been demonstrated here with semantic information of the environment based on ”complexity”, ”shape” and ”cluttering”. This information allowed this novel architecture to

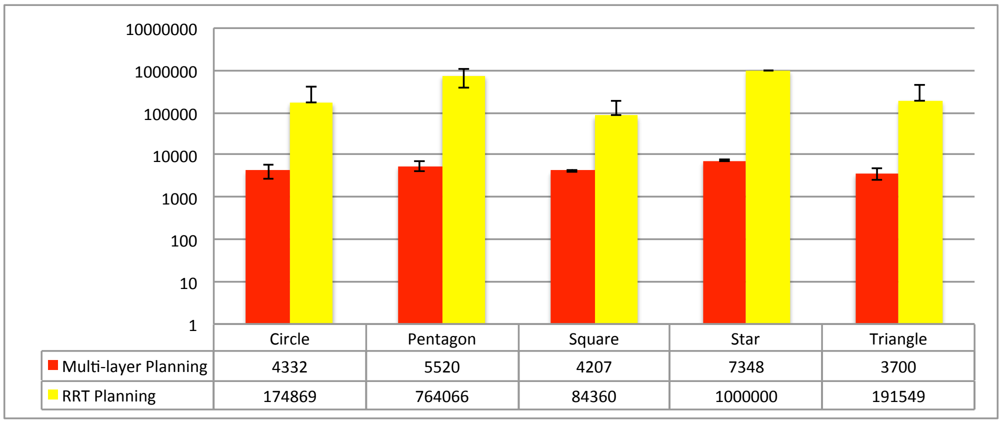


Figure 14: Average and standard deviation of number of random configurations used for path planning in the shapes application

deal efficiently with an abstract example using only simple geometrical path planning techniques.

However, real manipulation task for industrial processes involves more complex semantic information (functional surface, multi-physics interactions, surfaces or material properties). Future work will be done to further define both the meaningful semantic information needed for such tasks and the corresponding planning strategies. For instance, in assembly tasks, sliding motions are commonly used. We are planning to develop interactive geometric path planning methods with contact. We also plan to enrich the topological and semantic layer of our environment model in order to use our global architecture to plan paths with or without contact according to the functional context of the assembly tasks (or subtasks) to be performed. The proposed architecture meets the requirements for such semantic information.

Moreover, with an accurate semantic description, such a planner structure seems also well suited for off-line path planning allowing to rapidly find hard passages using the topological planning and to rapidly adapt the geometric planning strategy according to the local planning context.

References

[1] Aarno, D., Ekvall, S., Kragic, D.: Adaptive virtual fixtures for machine-assisted teleoperation tasks. In: International Conference on Robotics and Automation, pp. 1139–1144. IEEE (2005)

[2] Abbink, D.A., Mulder, M.: Neuromuscular analysis as a guideline in designing shared control. Advances in haptics **109**, 499–516 (2010)

[3] Ahmadi-Pajouh, M.A., Towhidkhah, F., Gharibzadeh, S., Mashhadimalek, M.: Path planning in the hippocampo-prefrontal cortex pathway: An adaptive model based receding horizon planner. Medical hypotheses **68**(6), 1411–1415 (2007)

[4] Anderson, S.J., Peters, S.C., Iagnemma, K., Overholt, J.: Semi-autonomous stability control and hazard avoidance for manned and unmanned ground vehicles. Tech. rep., DTIC Document (2010)

[5] Bordegoni, M., Cugini, U., Belluco, P., Aliverti, M.: Evaluation of a haptic-based interaction system for virtual manual assembly. In: VMR, Conférence on Virtual and Mixed Reality, pp. 303–312. Springer (July 2009, San Diego (USA))

[6] CGAL: CGAL, Computational Geometry Algorithms Library (2014). http://www.cgal.org

[7] Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., Thrun, S.: Princi-

ples of robot motion: theory, algorithms, and implementations (Intelligent robotics and autonomous agents). The MIT Press (2005)

[8] Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)

[9] Dragan, A.D., Srinivasa, S.S.: A policy blending formalism for shared control. International Journal of Robotics Research (2013)

[10] Fagg, A.H., Rosenstein, M., Platt, R., Grupen, R.A.: Extracting user intent in mixed initiative teleoperator control. In: American Institute of Aeronautics and Astronautics Intelligent Systems Technical Conference (2004)

[11] Fillatreau, P., Fourquet, J.Y., Le Bolloch, R., Cailhol, S., Datas, A., Puel, B.: Using virtual reality and 3d industrial numerical models for immersive interactive checklists. Computers in Industry (2013)

[12] Fiorentino, M., Radkowski, R., Stritzke, C., Uva, A.E., Monno, G.: Design review of cad assemblies using bimanual natural interface. International Journal on Interactive Design and Manufacturing (IJIDeM) **7**(4), 249–260 (2013)

[13] Flemisch, F.O., Heesen, M., Hesse, T., Kelsch, J., Schieben, A., Beller, J.: Towards a dynamic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations. Cognition, Technology & Work **14**(1), 3–18 (2012)

[14] Iacob, R., Mitrouchev, P., Léon, J.C.: Assembly simulation incorporating component mobility modelling based on functional surfaces. International Journal on Interactive Design and Manufacturing (IJIDeM) **5**(2), 119–132 (2011)

[15] Iacob, R., Popescu, D., Mitrouchev, P.: Assembly/disassembly analysis and modeling techniques: A review. Strojniski Vestnik/Journal of Mechanical Engineering **58**(11) (2012)

[16] Ladevèze, N., Fourquet, J.Y., Puel, B.: Interactive path planning for haptic assistance in assembly tasks. Computers & Graphics **34**(1), 17–25 (2010)

[17] LaValle, S.M.: Planning algorithms. Cambridge University Press (2006)

[18] Li, M., Okamura, A.M.: Recognition of operator motions for real-time assistance using virtual fixtures. In: Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS., pp. 125–131. IEEE (2003)

[19] Loizou, S.G., Kumar, V.: Mixed initiative control of autonomous vehicles. In: International Conference on Robotics and Automation, pp. 1431–1436. IEEE (2007)

[20] Lozano-Perez, T.: Spatial planning: A configuration space approach. Transactions on Computers **100**(2), 108–120 (1980)

[21] Marayong, P., Li, M., Okamura, A.M., Hager, G.D.: Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In: International Conference on Robotics and Automation, vol. 2, pp. 1954–1959. IEEE (2003)

[22] Taïx, M., Flavigné, D., Ferré, E.: Human interaction with motion planning algorithm. Journal of Intelligent & Robotic Systems **67**(3-4), 285–306 (2012)

[23] Weber, C., Nitsch, V., Unterhinninghofen, U., Farber, B., Buss, M.: Position and force augmentation in a telepresence system and their effects on perceived realism. In: EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics, pp. 226–231. IEEE (2009)

[24] You, E., Hauser, K.: Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input. Robotics: Science and Systems VII p. 354 (2012)

[25] Yu, W., Alqasemi, R., Dubey, R., Pernalete, N.: Telemanipulation assistance based on motion intention recognition. In: International Conference on Robotics and Automation, pp. 1121–1126. IEEE (2005)