H-RRT-C : Haptic Motion Planning with Contact

Nassime Blin, Michel Taïx, Philippe Fillatreau, Jean-Yves Fourquet

H-RRT-C : Haptic Motion Planning with Contact

Nassime Blin[1], Michel Taïx[2], Philippe Fillatreau[3] and Jean-Yves Fourquet[3]

***Abstract*— This paper focuses on interactive motion planning processes intended to assist a human operator when simulating industrial tasks in Virtual Reality. Such applications need motion planning on surfaces. We propose an original haptic path planning algorithm with contact, H-RRT-C, based on a RRT planner and a real-time interactive approach involving a haptic device for computer-operator authority sharing. Force feedback allows the human operator to keep contact consistently and provides the user with the feel of the contact, and the force applied by the operator on the haptic device is used to control the roadmap extension; on the contact surface, the orientation of the manipulated part is variable.**

Our approach has been validated through two experimental examples, and brings significant improvement over state of the art methods in both free and contact space to solve path-planning queries and contact operations such as insertion or sliding in highly constrained environments.

I. INTRODUCTION

Our goal is to perform industrial assembly/disassembly tasks in an immersive Virtual Reality (VR) environment by using a haptic device and automatic motion planning methods. In order to do this, we propose a novel approach based on the combination of haptic guidance forces and probabilistic path planning techniques.

Many works show the advantages of combining the use of a virtual reality environment and a haptic device providing force feedback. The authors of [1] assess the advantages of using both techniques simultaneously rather than separately, for education and training to manipulation tasks. [2] explains how interactive simulation using haptic feedback benefits from the cognitive and manual skills of the operator in the case of complex industrial assembly tasks. The authors address the interesting issue of the final insertion (low clearances assembly case), notably the imprecision of the contact simulation. In this case it is interesting to use a hybrid approach where traditional simulation of contact and motion are implemented for free movement under fixed assembly constraints, which are easy to integrate into the physics simulation solver.

In the robotics community, probabilistic motion planning techniques, such as RRT or PRM, have been intensively studied [3], [4]. Such methods are generic but can be very slow to solve problems in a highly cluttered environment (e.g. when dealing with a narrow passage or assembly tasks where the distance between the manipulated object and the obstacles are close to zero).

Several papers present interactive motion planning approaches featuring the use of motion planning techniques with a human operator in the loop [5], [6]. In [7], [8] the authors present a method for the cooperation between a human and an automatic motion planner. A haptic device can guide and improve the interaction between them [9], [10]. In other studies, the user interaction can be made using a haptically controlled object to modify or define critical object configurations, [11] [12] introduced an interactive planner using haptic force feedback. [13], [14] propose an interactive RRT-based path planning approach; an authority sharing paradigm allows the human operator to control the sampling process by using a haptic arm or a 6d mouse.

In our LGP laboratory, we adopted an original approach consisting in using simultaneously automatic motion planning techniques and interactive and immersive manipulation in virtual reality. It allows to take advantage of both the computational power of the computer involved and the higher cognitive capacities of the human operator. Automatic path planning techniques allow assistance to the manipulation. Using a haptic arm, force feedback allows to guide the operator along the computed trajectories. Control sharing is addressed as the operator may take the lead locally, for joint trajectory definition. We have first validated this approach in the free space, using purely geometrical models [15] [13] or higher abstraction level data (topology and semantics) for a better control of interactive path planning [16].

When simulating realistic industrial tasks, contact is needed. We have recently started works tackling the issue of interactive path planning with contact, without immersion first. In [17], we proposed a first interactive algorithm, based on a RRT algorithm, called I-RRT-C, and able to explore the whole workspace (free and contact spaces) and to plan directly on surfaces. An authority sharing parameter allowed to control the execution times dedicated to automatic path planning and to the capture of user defined configurations. Our approach brought promising improvements (more relevant paths obtained and drastic reduction of processing times, for better real time interaction) for path planning queries in highly cluttered environments [18]. The main limitations of our works at this stage were that our algorithms worked only at the contact of a single planar face, that parameters (notably the authority sharing scalar) were not adaptive, that the orientation of the manipulated part remained constant on the contact surface, and that the user intent was not being

[1]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France and Laboratoire Génie de Production (LGP), Université de Toulouse, INP-ENIT, Tarbes, France nblin_at_laas.fr

[2]LAAS-CNRS, Université. de Toulouse, CNRS, UPS, Toulouse, France taix_at_laas.fr

[3]Laboratoire Génie de Production (LGP), Université de Toulouse, INP-ENIT, Tarbes, France pfilatr_at_enit.fr, fourquet_at_enit.fr

This work was supported by Région Midi Pyréenées and Université Fédérale de Toulouse Midi Pyréenées and partly by ERC Grant 340050 Actanthrope

taken into account at the contact (as opposed to what was done in free space).

In this paper, we further develop this approach, but in the immersive context of our VR environment with haptic arm. Our main contribution is a new immersive and interactive path planning algorithm with contact called H-RRT-C. The following improvements are brought: a) force feedback allows the human operator to keep contact more consistently b) the force applied by the operator on the haptic device is used to control the roadmap extension and c) the orientation of the manipulated part is now variable.

This paper is organized as follows. Section II presents our interactive motion planning algorithm in contact, section III presents our multimodal interactive motion planner in free and contact spaces, section IV presents our software and hardware experimental architecture, section V presents our experimental results and section VI our conclusions and the next steps of our works.

II. Contact Haptic Path Planning

A. Interactive Motion Planning in Contact

The present work is based on our previous contribution, I-RRT-C [17] : Interactive Motion Planning in Contact without immersion. The aim of this RRT-based algorithm is to let the user cooperate with the computer. Using the speed of the computer and the cognitive capacities of the user, we observe significant improvement over using a standalone RRT.

The user is controlling the object to be planned in the workspace using a 6D mouse. The configuration of the object handled by the user is called q_{device}.

We introduced a parameter allowing to share authority between the computer and the user, called α. For each sample, a random number a is picked. If $a \leq \alpha$, a random configuration is shot and added to the roadmap if it is not in collision. If $a > \alpha$ the chosen configuration is q_{device}. By doing so, we can set given percentages of processing times allocated to the capture of human-defined and the computation of machine-defined input configurations.

The second contribution of I-RRT-C is contact sampling. When the user approaches an obstacle with the guided object, the planner switches to contact mode. The algorithm samples configurations on the surface the user approached with the object. This work suffered from many limitations, in particular we don't use user's intention, which will be improved in this new contribution with haptic arm. The haptic force feedback allows to take into account the user defined motion in contact to efficiently guide the extension of the roadmap.

B. Algorithm proposition

We have implemented a new algorithm based on I-RRT-C. Our novel algorithm, called H-RRT-C, is capable of interactive planning in contact while using our immersive architecture described section IV. We describe here only the contact subpart called $\boldsymbol{HapticSampling}$.

First, we improve contact sampling. Instead of sampling on a surface only bounded by the workspace limits given as an input to the planning problem, we now sample inside an ellipse $\mathcal{E}$ centered at the contact point. The ellipse parameters depend on the user's intention. Samples are constrained around a user-defined point; this allows to drastically reduce colliding points or irrelevant samples not consistent with the user's sampling intention.

Second, we implemented a method changing the sampling behavior in function of the actions of the user. When getting in collision with an obstacle, the user feels force feedback, thanks to a haptic arm and a collision detection algorithm provided by the haptic arm constructor. Thus, we can use the force provided by the operator through the device at any time. We can also measure the position and orientation of the moving object handled with the haptic arm.

This algorithm adapts contact sampling using the intentions of the operator by measuring three parameters given through the interactive device.

1) The intensity of the colliding user force $\|\boldsymbol{f_u}\|$: it traduces the interest of the operator for the current contact surface. For example, when the operator slightly touches a surface then goes away, this may be either by mistake or intentionally. On the opposite, if a user pushes firmly the manipulated object towards a surface, we consider this is done intentionally. The surface of $\mathcal{E}$ is made proportional to the force applied by the operator.

2) The angle to normal φ: it expresses that the operator intends to move. If this angle φ to the normal to the contact surface $\boldsymbol{n_c}$ is equal to zero (i.e. the force applied by the user is perpendicular to the contact surface), we have no information about any intention and sample inside a disk. This normal to contact is obtained using Gram-Schmidt algorithm as used in I-RRT-C [17] If the operator lets the manipulated object slide along the contact surface, the measure of φ allows to determine his intentions and the algorithm adapts by sampling more along one direction by elongating the ellipse, see figures 1 and 2.

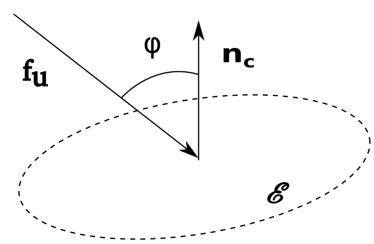

Fig. 1: Tangent force, big φ.

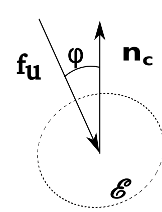

Fig. 2: Small φ.

3) The Δ_p vector: : it indicates towards where the user is actually moving. Its projection on the ellipse is Δ_t The big axis of $\mathcal{E}$ is aligned with this vector's projection on the contact surface. This permits to sample in the preferred direction, see figure 3.

C. Haptic sampling algorithm details

We present here the details of our $\boldsymbol{HapticSampling}$ algorithm using the haptic force feedback. This algorithm is called whenever the handled object collides with an obstacle of the environment W.

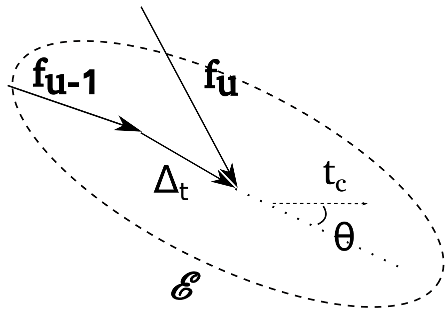

Fig. 3: User movement aligns ellipse.

Algorithm 1 HapticSampling

Require: W
1: **if** contact **then**
2: $\boldsymbol{f_u} \leftarrow$ HapticArm
3: $\boldsymbol{n_c} \leftarrow$ HapticArm
4: $\boldsymbol{\Delta_p} \leftarrow$ HapticArm
5: $\varphi = angle(\boldsymbol{f_u}, \boldsymbol{n_c})$
6: $\theta = angle(\boldsymbol{\Delta_p}, \boldsymbol{t_c})$
7: $q_{current} \leftarrow$ContactShooter($\varphi, \theta, \boldsymbol{f_u}$)
8: return $q_{current}$
9: **end if**

In lines 2, 3 and 4, we read the information given by the haptic arm. With the haptic arm the user produces a 3D force $\boldsymbol{f_u}$ when the handled object touches an obstacle in the environment. This force quickly grows when in contact because of collision detection and force feedback. The operator can push firmly against obstacles, thus generating a large force $\boldsymbol{f_u}$ (for example in the case of an insertion task).

When the user collides with an obstacle surface, we get the first colliding point, and compute the normal $\boldsymbol{n_c}$ to the contact surface at this point.

While being in contact, we compute the $\boldsymbol{\Delta_p}$ vector formed by the subtraction of two consecutive positions. $\boldsymbol{\Delta_p}$ corresponds to the motion intended by the user.

With these different information we can define the characteristics of the sampling ellipse $\mathcal{E}$.

- the surface of the the ellipse in which configurations will be shot depends on $\|\boldsymbol{f_u}\|$,
- φ defines the ratio between the dimensions of the two axes (line5),
- θ defines the orientation of the ellipse (line 6)

We compute φ as followed:

$$\varphi = acos\frac{\boldsymbol{f_u}.\boldsymbol{n_c}}{\|\boldsymbol{f_u}\| * \|\boldsymbol{n_c}\|} \tag{1}$$

If this angle is zero, the small and big axis of the ellipse are the same length and the ellipse becomes a disk. Whenever this angle grows, the ellipse becomes more and more elongated; the length of the major axis increases and the length of the minor axis decreases. Let s_a an b_a be the length of the minor axis and the major axis respectively. We compute the axis lengths as follows:

$$b_a = exp(2 * \varphi) \qquad s_a = 1/b_a \tag{2}$$

We use the tangent vector to the surface $\boldsymbol{t_c}$ and $\boldsymbol{\Delta_p}$ to compute the orientation θ of the ellipse. First $\boldsymbol{\Delta}_p$ is projected on the sampling surface : $\boldsymbol{\Delta}_t = \boldsymbol{\Delta_p} - \boldsymbol{\Delta_p}.\boldsymbol{n_c} * \boldsymbol{n_c}$

Then, we compute:

$$\theta = acos\frac{\boldsymbol{\Delta}_t.\boldsymbol{t_c}}{\|\boldsymbol{\Delta}_t\| * \|\boldsymbol{t_c}\|} \tag{3}$$

Now we can randomly draw new configurations $q_{current}$ (line 7) to be sampled inside the surface defined by the ellipse.

D. Example

The following example presents the way the three parameters are used by sampling on a planar surface. Various situations are presented to describe the use of every parameter. For each example, user is pushing a cube against the green wall.

Figure 4 presents an example where the user pushes gently towards the green surface. Configurations are sampled in a small area around the contact point.

Figure 5 represents the results obtained when the user pushes stronger. The figure shows a first small ellipse surrounded by a much bigger one. Borders are displayed using dashed lines for clarity. The user pushed slightly then pushed more firmly. These two actions gave two ellipse sizes. Figure 6 shows the ellipses in perspective.

Figure 7 present a case where the user pushed the cube orthogonally to the surface. We get a round shape. Next is figure 8 where we see a very elongated ellipse, this is because there is a big φ angle between the user force $\boldsymbol{f_u}$ and the normal to the surface $\boldsymbol{n}$.

Figure 9 presents a last exemple with two behaviors whose corresponding ellipses are surrounded by dashes for display clarity reasons. At the center we can see an almost round large ellipse surrounded by a second very large ellipse. The latter is due to a very large $\boldsymbol{f_u}$, which leads to sampling very far away from the contact surface.

III. Interactive Haptic Path Planning

Using the $\boldsymbol{HapticSampling}$ algorithm in an interactive approach we can plan trajectories for an object manipulated in a highly constrained environment and achieve tasks such as sliding and insertion by using force feedback for better control on (faster) contact sampling.

Our new algorithm permits authority sharing when sampling in contact which was not allowed in I-RRT-C algorithm because contact sampling on the contact surface was completely automatic. Now we use the intentions of the operator during contact with three different parameters to speedup the overall process.

Moreover, the orientation can be easily changed during contact by using the force applied by the operator, which was not allowed in our previous works. This allows the operator to change orientation in real time which greatly simplifies his task.

Contact mode starts when at least one collision is detected. We get out from contact mode as soon as no more collisions

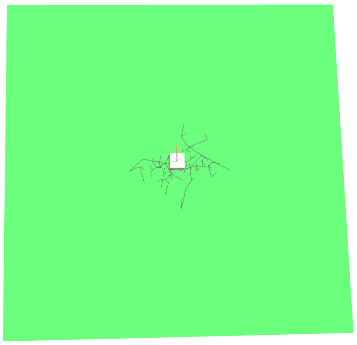

Fig. 4: Small force.

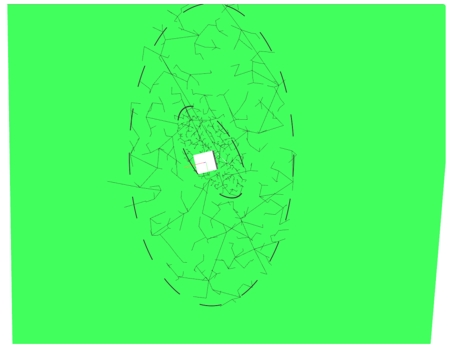

Fig. 5: Big and bigger force.

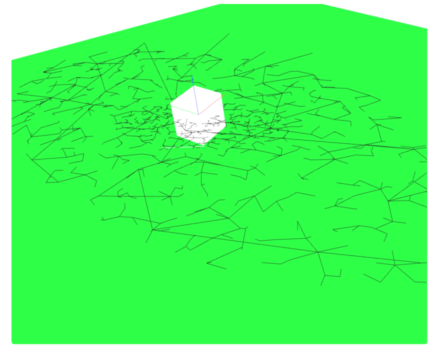

Fig. 6: Same ellipses in perspective.

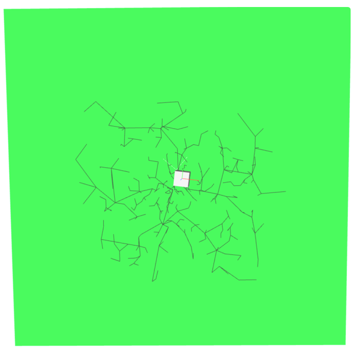

Fig. 7: Force normal to surface. Ellipse has a round shape.

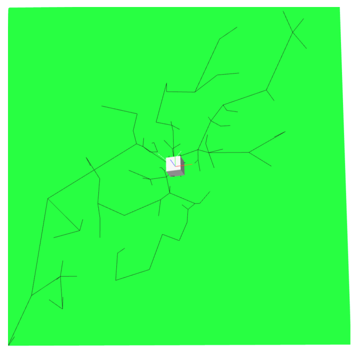

Fig. 8: Big angle φ between force and surface. Elongated shape.

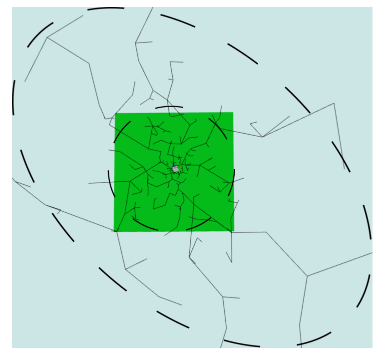

Fig. 9: High force, round then very high force, orientated.

Algorithm 2 H-RRT-C: Haptic RRT in Contact

Require: $W, T, HapticArm, \alpha$

1: **loop**
2:   **if** contact **then**
3:     $q_{current} \leftarrow$HapticSampling(HapticArm)
4:     $T \leftarrow$ Add_Tree($q_{current}$)
5:   **else**
6:     $a \leftarrow$ rand$(0, 1)$
7:     **if** $a \leq \alpha$ **then**
8:       $q_{current} \leftarrow$ RandomShooter()
9:       $T \leftarrow$ Add_Tree($q_{current}$)
10:     **else**
11:       $q_{current} \leftarrow q_{device}$
12:       $T \leftarrow$ Add_Tree($q_{current}$)
13:     **end if**
14:   **end if**
15: **end loop**

are detected. We choose the sampling surface as the plane defined by the normal to the first contact.

Line 3, we call $\boldsymbol{HapticSampling}$ described in algorithm 2. It adds a valid contact configuration to the tree (line 4) using the operator advises.

Line 5, the algorithm is not in contact mode. A random number a between 0 an 1 is shot, line 6

Line 7 we choose human or machine mode. If $a \leq \alpha$ we are in machine mode otherwise we enter human mode (line 10). The α parameter is a fixed number describing the percentage of machine input configurations.

Line 8 is the probabilistic motion planning mode where a completly random configuration is shot and added to the tree if valid.

Line 11 is the human mode. The current configuration $q_{current}$ of the object held by the operator in the free space is added to the tree T.

Our examples show that the sampling surface in contact can be changed in real time with three parameters θ, φ and $\boldsymbol{\Delta}_t$ given by the operator. The shape of the ellipse traduces his intention. The instantaneous movement is the direction in which the object should move. The applied force is the interest the user have in the current surface.

IV. ARCHITECTURE

Our last contribution is an architecture that permits interactive motion planning using the LAAS-Gepetto motion planner: HPP [19], along with a virtual reality (VR) environment provided by LGP-ENIT laboratory.

A. Overview

The VR environment uses a haptic arm Virtuose 6D 35-45. It is a sensorimotor bidirectional device. We also have a large 3D screen and a motion capture system attached to the 3D glasses.

Two separate simulations are run simultaneously, on two separate computers. Motion planning is run with HPP [19] embedding our H-RRT-C algorithm that starts motion planning threads. It runs on a first computer under Ubuntu 14.04.

The second computer runs Virtools under Windows XP controlling the RV environment the user will interact with.

Both computers use the same mesh models of the environment and the robot. We use ZeroMQ [20] to communicate through the network between these two different computers.

The user moves the arm whose configuration q_{device} in the workspace (the virtual environment) is sent to Virtools. Visualization of the process and moving around the scene is also done in Virtools using our virtual reality hardware : a man sized 3D screen, 3D glasses and markers attached to it. Last, a 6DoF robotic arm is used as a haptic device. The figure 10 shows the general architecture.

B. Motion Planning

Hpp software handles planning process. Our implementation uses three different threads : called InteractivePlanner, OperatorThread and DeviceThread, see figure 11.

OperatorThread reads the operator configuration variable q_{device} given by the interactive device. It also lists obstacles, computes normals and distances to the object. Last, it triggers contact mode.

DeviceThread opens the interactive hardware : 6D mouse or Haption haptic arm. It sets the operator configuration variable q_{device}. When the haptic arm is used, it gets the data through 0mq from Virtools.

InteractivePlanner is a derived class from a BiRRT algorithm implemented in Hpp. It is our implementation of IRRTC and runs motion planning while managing interaction and contact. It chooses the operating mode : random, operator or contact (see section II).

C. Virtual reality features

As our goal is an immersive experience, we have implemented virtual reality features to help the user to feel contact with the virtual environment. We believe that immersive motion planning will help the operator to have a better understanding of his environment and thus, perform tasks more naturally and more accurately. Our previous work used a 6D mouse as an interaction device. Though we had a human control on motion planning, we had no sense of environment, and we had no immersion capabilities.

The first feature is the replacement of the 6D mouse by a user-controlled haptic arm model Virtuose 6D by Haption company, see figure 13. As before, the operator can move an object in six dimensions. The difference is that we now have a physicalisation process of the environment and object mesh models to a voxmap consisting of unbalanced voxels. This is done by a library provided by the haptic arm constructor and is used in our virtual reality software, virtools. Collisions of the object with the environment are now felt by the operator through the haptic arm. Force feedback permits the user to feel obstacles.

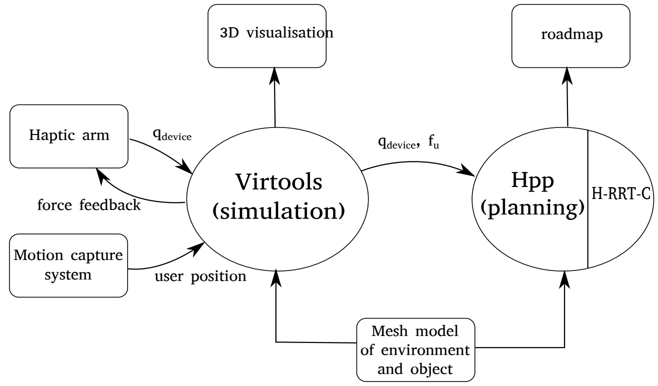

Fig. 10: Overview of the architecture.

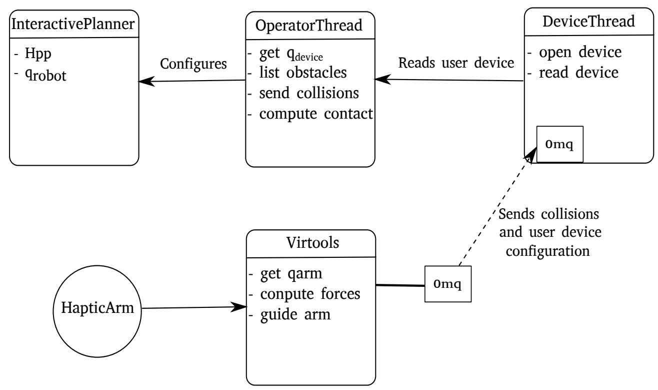

Fig. 11: Architecture of the haptic planner.

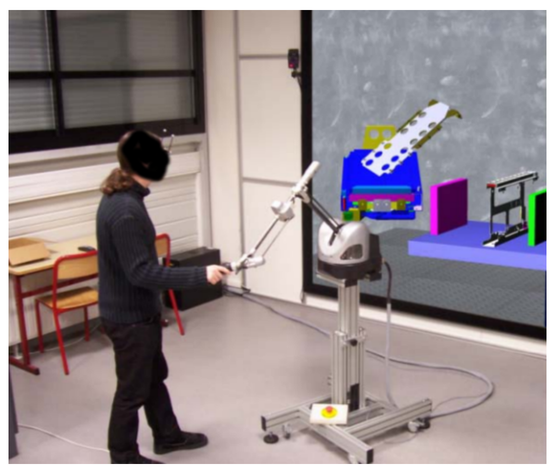

Fig. 12: Virtual reality environment

Fig. 13: 3d glasses with markers.

Second, we have a large 3D screen with 3D glasses. Letting the user see the scene in three dimensions can help him to better interpret the available space for moving around.

Third we have a motion capture system and we have markers fixed on the 3D glasses, see figure 13. The motion capture can then capture the movements of the head of the operator and change the viewpoint of the virtual scene according to his actions. The operator can then move around the scene or zoom in or out simply by doing the appropriate moves.

V. Experimental results

We have tested our H-RRT-C algorithm on various environments. These are the same we used on our previous work [17] [18] to be able to compare algorithm performance and results easily.

A. Narrow tunnels

We first tested our algorithm on an environment that consists in a long tube inside which the object can barely move except for crossing this very narrow passage, see figure 14. The goal position is displayed on the left of the figure. It is followed by two thin planes with a small opening. The first plane is rotated around one axis, the second rotated around two axis. This makes a very cluttered place and a big challenge for an operator as the object needs to be rotated correctly and very accurately. The result is shown figures 15 for an average α values.

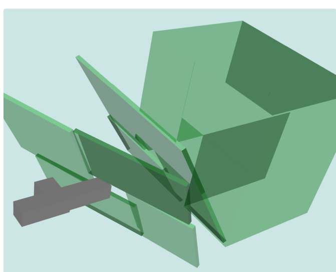

Fig. 14: Narrow tunnels.

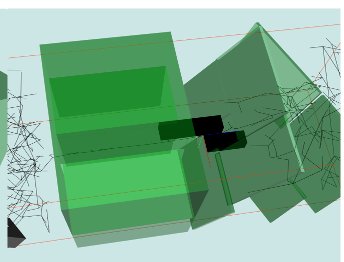

Fig. 15: alpha = 0.5

α	Time (s)	Nodes	Edges
0	135	2 110	4 218
0.05	60	871	1 740
0.1	67	924	1 846
0.2	27	536	1 070
0.5	33	661	1 320
0.8	28	671	1 340
0.95	38	669	1 336

TABLE I: Influence of α with H-RRT-C.

The table I shows the results for various α values. Extreme values considerably degrades the algorithm performances, needing more than two minutes for a scenario without any computer help $\alpha = 0$ to only 27 seconds with $\alpha = 0.2$. These results should be compared to our previous best performance of 61 seconds using I-RRT-C best parameters [18]. We solve the problem twice faster.

B. Crossing planes

The second environment is composed of two pairs of planes in which the free space is narrow. This environment only slightly differs from the one used in our previous work, see [17]. Figure 16 show the result with a zero α value. Figure 17 show the nodes samples at the surface of one of the first two planes as the user pushed the object against this surface.

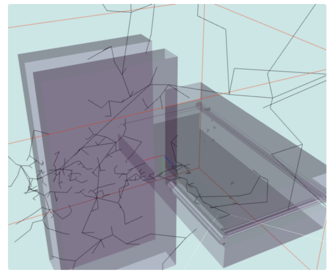

Fig. 16: alpha = 0

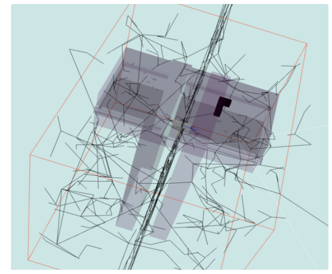

Fig. 17: alpha = 0.1

Scenario	α	Time (s)	Nodes	Edges
RRT	1	2h13m57s	6 919	13 836
I-RRT-C	0	25	961	1 920
Haptic I-RRT-C	0	17	1 105	2 202

TABLE II: Influence of algorithm, second scenario.

This environment is tested with various algorithms, results in table II. These test do not measure H-RRT-C but previous methods. In the first line we try to solve the with a standard RRT. This method takes more than two hours to find a solution.

We want to compare our performances to our previous work I-RRT-C so we measured how much time is needed by an experimented operator to solve the problem using this algorithm and a 6D mouse. We choose alpha to be zero meaning that no free space random nodes are shot but we still have contact nodes generated when in contact mode. The problem is solved in 25 seconds.

For the third test we chose Haptic I-RRT-C algorithm. It is I-RRT-C but only replacing the mouse by a haptic arm. It improves performances by 50% reducing the time to as low as 17 seconds.

Scenario	α	Time (s)	Nodes	Edges
H-RRT-C	0	15	484	966
H-RRT-C	0.005	18	778	1 548
H-RRT-C	0.01	10	669	1 336
H-RRT-C	0.1	12	862	1 722
H-RRT-C	0.3	20	926	1 850
H-RRT-C	0.6	21	954	1 897
H-RRT-C	0.9	39	1 262	2 522

TABLE III: Influence of α, second scenario.

The H-RRT-C algorithm is then benchmarked with several tests using different α values, see table III. Apart from very high values, we are almost always faster than the best I-RRT-C parametrization [17]. Optimal α values should be around 0.1 with an experiment duration as low as 10 seconds which is much faster than any other method.

C. Discussion

Our algorithm is capable of solving motion planning queries using the capabilities of both a computer and a human operator.

From the results shown above, we can see that for both environments, we solve the problems 50-200% faster than our previous algorithm depending on the geometry of the problem.

The second experiment shows that the sole use of a haptic arm instead of a non-actuated 6D mouse speeds up the process. It also gives the possibility to an untrained operator to work unlike the 6D mouse which needs training.

Our tests show that our new sampling method also contributes to speeding up the overall method. It is done by constraining contact sampling around the user's movements and by generating them by taking into account its intentions.

Regarding the α parameter, our previous work I-RRT-C was very dependent on a good suited α value. The time needed to solve the problem had big variations depending on its value. With our new work H-RRT-C, we observe that we can choose any α apart from extreme values and the problems would still be solved twice quicker using our new method.

VI. CONCLUSION

In this paper, we have proposed a novel haptic path planning algorithm with contact, called H-RRT-C, allowing to explore both the free and contact spaces. When planning in contact, several significant improvements are brought. The feel of the contact through force feedback allows the human operator to keep contact more consistently. The force applied by the operator on the haptic device is used to take the user intent into account and to constrain node sampling in a user defined area.

In our future works, we will address the issue of adapting the algorithm parameters such as ellipse size and interactivity parameter in real time in function of the user's behavior. Regarding contact we will develop a method that automatically switches contact planning on successive surfaces instead of having to choose them manually. On the other hand, we also intend to plan simultaneously on several surfaces to move at the contact of complex surfaces.

REFERENCES

[1] D. Grajewski, F. Grski, A. Hamrol, and P. Zawadzki, "Immersive and haptic educational simulations of assembly workplace conditions," in *Inter. Conf. on Virtual and Augmented Reality in Education*, 2015.

[2] J. Perret, C. Kneschke, J. Vance, and G. Dumont, "Interactive assembly simulation with haptic feedback," *Assembly Automation*, vol. 33, no. 3, 2013.

[3] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, 1996.

[4] J. Kuffner Jr and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE Int. Conf. on Robotics and Automation*, 2000.

[5] J. Denny, J. Colbert, Q. Hongsen, and N. Amato, "On the theory of user-guided planning," in *IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[6] J. Denny, "Collaborative motion planning," in *Ph.D. Thesis, Department of Computer Science and Engineering, Texas A&M University*, Aug 2016.

[7] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," *Autonomous Robots*, 2001.

[8] Y. Yan, E. Poirson, and F. Bennis, "Integrating user to minimize assembly path planning time in plm," in *Product Lifecycle Management for Society*. Springer, 2013, pp. 471–480.

[9] J. Rosell, C. Vázquez, A. Pérez, and P. Iñiguez, "Motion planning for haptic guidance," *Journal of Intelligent and Robotic Systems*, 2008.

[10] C. Vzquez, J. Rosell, L. Chirinos, and O. A. Domnguez, "Haptic primitives guidance based on the kautham path planner," in *IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems*, 2010.

[11] X. He and Y. Chen, "Haptic-aided robot path planning based on virtual tele-operation," *Robotics and computer-integrated manufacturing*, vol. 25, no. 4, pp. 792–803, 2009.

[12] N. Ladeveze, J. Y. Fourquet, B. Puel, and M. Taix, "Haptic assembly and disassembly task assistance using interactive path planning," in *Virtual Reality Conf.* IEEE, 2009, pp. 19–25.

[13] D. Flavigné, M. Taïx, and E. Ferré, "Interactive motion planning for assembly tasks," in *The 18th IEEE International Symposium on Robot and Human Interactive Communication. RO-MAN 2009.* IEEE, 2009.

[14] M. Taïx, D. Flavigné, and E. Ferré, "Human-robot interaction throught motion planning algorithm," *Journal of Intelligent and Robotic Systems*, vol. 67, no. 3, 2012.

[15] N. Ladevèze and J.-Y. Fourquet, "On the collaboration of an automatic path-planner and a human user for path-finding in virtual industrial scenes," in *Inter. Conf. on Control Automation Robotics & Vision (ICARCV)*, Singapore, 2010.

[16] S. Cailhol, P. Fillatreau, J.-Y. Fourquet, and Y. Zhao, "A hierarchic approach for path planning in virtual reality," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 9, no. 4, pp. 291–302, 2015.

[17] N. Blin, M. Taïx, P. Fillatreau, and J. Y. Fourquet, "I-rrt-c: Interactive motion planning with contact," 2016.

[18] ——, "Tuning interaction in motion planning with contact," in *IEEE Inter. Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2016.

[19] J. Mirabel, S. Tonneau, P. Fernbach, A. K. Seppl, M. Campana, N. Mansard, and F. Lamiraux, "Hpp: A new software for constrained motion planning," in *IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[20] Zeromq. [Online]. Available: http://zeromq.org/