Int J Interact Des Manuf (2010) 4:95–102
DOI 10.1007/s12008-010-0091-7

ORIGINAL PAPER

Interactive simulation of CAD models assemblies using virtual constraint guidance

A new method to assist haptic assembly tasks

Loïc Tching · Georges Dumont · Jérôme Perret

Received: 30 June 2009 / Accepted: 8 March 2010 / Published online: 24 April 2010

Abstract In the context of virtual reality (VR) and of computed aided design (CAD), haptic simulations are used to perform assembly tasks between 3D objects. To ensure the good assembly of those objects, we propose a new method of interactive assembly that uses both kinematic constraints and guiding virtual fixtures. Modelling a haptic assembly task as a combination of mechanical joints, we focus on the guidance of objects and on the activation cues of kinematic constraints in physical simulation. In this article, we first outline the difficulties related to the haptic-assembly of CAD objects in VR simulation. Introducing the virtual constraint guidance (VCG), we present a new method for haptic guidance that decomposes a task in two independent steps: a guiding step which use geometries as virtual fixtures to position objects, and a functional step which use kinematic constraints to perform the assembly task. We finally present a complete application of our method on a peg-in-hole insertion task.

Keywords Virtual reality · Haptic simulation · 3D assembling task · Virtual fixture · Kinematic constraint · Geometric guidance · Peg-in-hole

1 Introduction

Within the industrial framework, applications that need haptic interfaces mainly consist in combining assemblies and disassemblies tasks to valid virtual prototypes or to train operators. In such cases, the traditional use of haptic is to control a CAD object and let it interact within virtual reality environments. To assist users in performing the assembly of CAD objects, the fields of teleoperation and virtual reality provide methods resulting from algorithmic assembly planning [1]. The objective of such assembly planning is to determine sequences to assemble a product from its individual parts, ensuring that the moving objects do not collide with any other objects in the working environment. Nevertheless, in complex assembly simulation, the user can intuitively interact with the virtual CAD environment. While solving various path planning problems, he can find realistic solution that algorithmic approaches can not find. The human expertise of mechanical assembly can be extracted and used for the automatic generation of interactive assembly sequences [2]. It is then possible, by using virtual fixtures, to assist the user while leaving him a partial control of his movements. Introduced by Rosenberg [3], virtual fixtures are abstract perceptual information added to the simulation that help user to perform specific tasks.

In this article, we present a method that combine virtual fixtures and kinematic constraints to perform assembly tasks with haptic interaction. The strong point of our approach is to allow the direct use of CAD data in virtual reality (VR) environment by introducing the concept of virtual constraint guidance (VCG). This guidance method applies local constraints at the place of the assembly, which force objects to move along pre-defined paths. The main idea is to use both a physically-based exploration of the VR environment, and

---

L. Tching (✉)
Haption, IRISA Bunraku (INRIA joint team),
Rennes, France
e-mail: loic.tching@haption.com

G. Dumont
ENS Cachan, IRISA Bunraku (INRIA joint team),
Rennes, France
e-mail: georges.dumont@bretagne.ens-cachan.fr

J. Perret
Haption, Soulgé-sur-Ouette, France
e-mail: jerome.perret@haption.com

a constraint-based execution of assembly tasks, while deactivating locally the collision detection.

This paper is organized as follows: Sect. 2 presents the context and previous work of haptic assembly of CAD models, focusing on the geometry approximations induced by the CAD-VR data exchange and by collision detection algorithms. Section 3 introduces the concept of virtual constraint guidance, our generic approach for the assembly of CAD objects in VR environment. We detail the use of virtual fixtures and kinematic constraints in the interactive simulation in order to perform assembly tasks. In Sect. 4, we illustrate our concept on a basic insertion task (example of peg-in-hole) and its application to an industrial case. Section 5 presents an evaluation of our guidance method. We conclude in Sect. 6 and propose future works.

2 Context and related work

2.1 Limits of geometric models

A field of Computer Aided Design (CAD) concerns the design, the handle and the assembly of 3D objects that represent manufactured products. In projects where the design is done by different companies, the digital mock-ups issued from different sources may induce format incompatibility, then forbidding simple assembly sequence. Indeed, the complex process of designing involves data that have been created by different software and for several purposes. This problem is all the more important when such different parts of an assembly are weld together into a VR application.

A major limit using CAD models in VR application is related to the exchange of data between the applications, which implies geometric approximations of the original CAD models. At the time of the design, CAD objects consist of a precise model description containing analytic, topology and dimension information. Those information are needed for manufacturing process or for computational purpose. While the surfaces of the models are described by nurbs or splines in CAD environment, they have to be tessellated to be used in VR simulation. This conversion is done by the export filter of CAD softwares, which generate polygon-based descriptions of the models. The discretization of the original continuous representations leads to parts that are geometrically incompatible with each other. During the conversion process, a component may not be tessellated enough so that it may lost important geometric details. The difficulty to control such backlashes leads to difficult adjusting of the assemblies. For example, at the time of its CAD design, a peg is represented by a circle-based prism associated to topological information of cylinder. On the left part of Fig. 1, it can be inserted in a corresponding hole, whereas it is no longer possible once its geometry has been converted and re-used in a VR application (right part of Fig. 1).

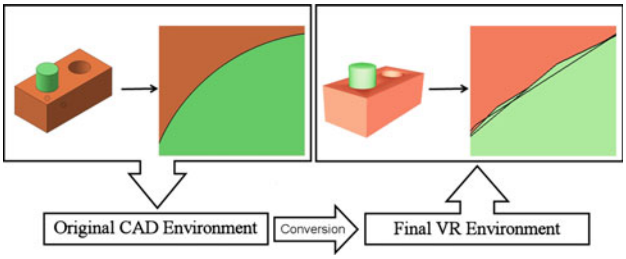

Fig. 1 Loss of geometry accuracy between CAD and VR applications, due to conversion

2.2 Limits of collision detection models

To assist the user in performing an assembly task, we consider that the user gets a classical haptic control of an object. The physical simulation follows a traditional use of haptic, which consists in controlling the degree of freedom of an object and return force feedback to the user thanks to a physical simulation process. In simulators that are based on physical models, positions and velocities of the simulated objects are calculated by the integration of the differential equations of dynamics. This solid mechanics approach allows to describe the solids evolutions in time. The physical computation of force feedback could be performed by non smooth dynamics methods.

The time-stepping approach of non-smooth dynamics takes into account the discontinuities of velocities during the contact phase, by computing the average interaction forces on a fixed time step, then by reintroducing it in a formulation of the dynamics equations. The non-smooth dynamics deals with complementary problems, i.e., which formalize the relation between points relative positions and Lagrange multipliers, which represent the contact forces. Used on systems that deal with non-smooth laws of interaction (discontinuous functions) it allows to manage unilateral treatments of local interactions such as contact, friction and adhesion. This approach, described in [4] is used in the cases of collisions between solids, where relative speeds adopt discontinuous behaviors and prohibit the calculation of the objects' positions by integrating the equations of dynamics. The contact between objects can be described by physical models to simulate the objects behavior in a realistic way. The treatment of the interactions is issued of the contact information delivered by the detection step. In order to provide collision information, there is a vast field of research on collision detection, and multiple existing paradigms. Kockara et al. [5] draw up surveys of the various approaches of detection between polyhedrons.

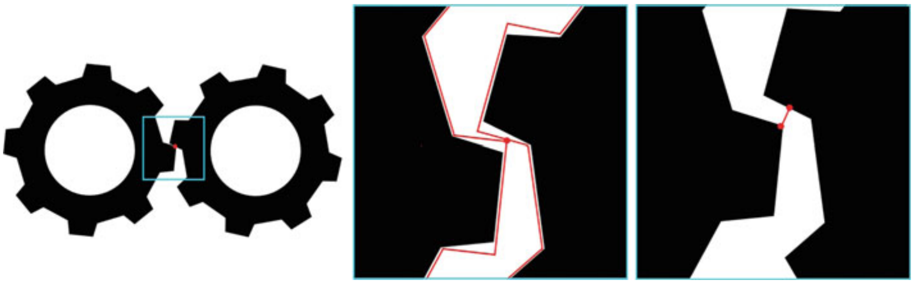

Fig. 2 Two different methods that compute the collision place between two gears

For assemblies, a major limit is related to the collision detection used in VR session. In haptic simulations and physical interactive simulation more generally, it is necessary to detect, in real time, the collisions between objects. Most of detection methods turn the original geometric representation into a simple one in order to accelerate the detection process, or to make them compatible with the physical engines. This lead to geometrical approximations of the objects involved in assemblies. The size or the form of the resulting geometrical models differs from the original objects so that the assembly backlashes are no more insured. In haptic simulation, most of detection method decompose the detection process in several phases, which are linked to the objects proximity: the narrow and broad phases [6]. Figure 2 illustrates two detection methods that compute non trivial position of the local contact point between two colliding gears. The first method, based on bounding volumes [7] (down-left picture), locates the collision at the intersection of bounding volumes. The second method, based on local minimum distance [8] (down-right picture), defines the collision as a threshold distance between geometries. Those two examples induce geometric approximation that forbid the objects to collide on their exact geometries. The resulting gaps thus prevent the geometries to be assembled as the initial plan.

These two factors show the limit of the use of CAD objects in VR applications. The tessellation of the objects introduces backlashes and approximation of the original geometries, while collision detection adds extra imprecision in the assemblies. To avoid the problem of geometry conversion, solutions like VADE [9] directly exchange geometries and trajectories information with the CAD environment, and then create kinematic motions and constraints during virtual assembly. Bourdot et al. [10] integrated VR and CAD in a common framework making possible intuitive and implicit edition on 3D objects within Virtual Environments. Gupta et al. [11] demonstrate that force feedback through the haptic interfaces reduced the time needed for assembly completion and reduced the number of handling errors occurring during the task. For this, they have developed a desktop system called the virtual environment for design assembly (VEDA) that can uses force feedback, physically based modeling, sound cues, and stereo vision to provide a more realistic virtual assembly experience. A limit of this kind of dual approach is that it does not allow to use the CAD object directly in VR simulation. To achieve this, it is still necessary to export the 3D data in VR compatible format and modify the geometries to fit the assemblies. The method we propose in next section avoids the problems of tessellated geometries and collision detection. By preserving assembly properties issued from CAD model, this method directly uses the object imported from CAD application.

Seth et al. [12] uses physics-based modeling for simulating realistic part behavior and provides a dual-handed haptic interface for mechanical assembly in an immersive VR environment. They proposes a modeling for assembly of B-Rep objects: as part of the SHARP platform, they study the influence of collision detection combined with assembly constraints in haptic insertion of a peg-in-hole. For example the insertion of a cylinder (peg) in a hole of the same diameter could basically be considered a slider-pin joint (cylindrical joint) between the two objects.

3 Virtual constraint guidance

We propose a generic approach for the assembly of CAD objects in VR environment and introduce the concept of virtual constraint guidance (VCG) for insertion tasks. The main idea of this concept is to combine virtual fixtures that guide the object to specific position, with mechanical joints that limit the relative movement possibilities. The specificity of this new approach is that the handled objects are directly imported from CAD applications without any modification of their meshes. They are subjected to classical physic law for haptic rendering, whereas we propose to disable collision detection during the insertion task. For that, we firstly consider an assembly task as the sum of elementary tasks, which could be modelled as mechanical linkages (prismatic, ball linkage, hinge …). For example the insertion of a cylinder (peg) in a hole could be considered a slider-pin joint (cylindrical joint) between the two objects.

3.1 Concept: assembly as a combination of mechanical linkage

The area related to the assembly tasks is the functional area where the user is supposed to perform the assembly and where it is then possible to assist the realization of the task. The motivations of this assistance could be linked to the problems mentioned in previous section or linked to the assembly complexity. On one hand, the user may not have a priori knowledge of the final assembly. On the other hand, the assembly may not propose any intuitive composition and need unusual or complex movements.

To apply the constraint based guidance in the functional area, the first step consists in qualifying the task and associating both mechanical linkage(s) and virtual fixture(s). In practice, the mechanical linkage is modelled as a kinematic constraint and the virtual fixture is modelled as complex virtual walls, geometric guides that limit the movements of the user. To create the constraints, we use topological information of CAD objects to identify the assembly areas, and model the trajectory with mechanical linkages, when possible. This phase is carried out by pre-processing and placing constraints for each zone of interest. It could be done manually thanks to CAD information of assemblies.

Once the constraints and their associated geometric guides are set-up, the simulation switches between two different modes of control, related to the two areas described above.

3.2 Positioning of the object

To activate the kinematic constraints in the simulation, it is preliminary needed to place the objects in a spatial configuration where the constraint, and only the constraint, will guide the user. For that we propose to match geometric attributes of the objects and the constraint; it is then possible to insure the correct positioning by only insuring a spatial correspondence between geometries. For example, to activate a slider-pin joint between a peg and a hole of the same diameter, it is just needed to align the axis of the peg with the axis of the constraint.

Figure 3 illustrates two examples of peg-in-hole positioning. On the left case, the positioning of the cylinder over the hole is equivalent to the alignment of two segments (or two pairs of point): the coincidence of the cylinder axis and the joint axis. On the right case, the positioning of the prism is equivalent to the alignment of three edges (or three pairs of point).

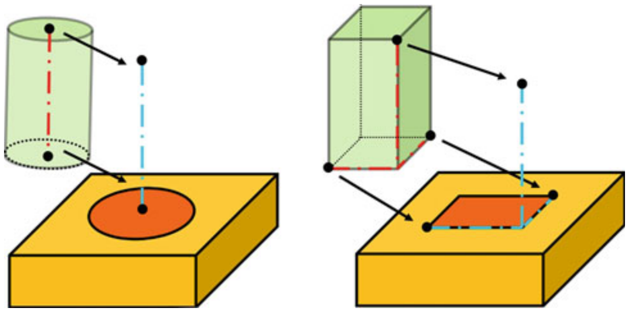

Fig. 3 Geometric requirements to activate a mechanical constraint

To help the user in positioning the objects, we propose the creation and the use of virtual fixtures in the simulation. The virtual fixture we use are abstract perceptual information added to the simulation that help user to perform specific task. Such class of guidance modes, which are implemented in high-level software, limits the user movement into restricted regions or influences its movement along desired paths. Within our framework, we use virtual walls that are implementation of virtual fixtures. Those abstract impedance surfaces, we named geometric guides, are added to the scene and provide visuo-haptic assistance.

For basic assembly cases, we consider two groups of geometric guides: the moveable ones that are linked to the driven object, and the fixed ones which are linked to the fixed object of the assembly. These geometric guides interact each other in terms of collision detection, but can not interact with the geometries of the objects. This leads to different combinations of contact:

- contact between geometric guides, which are indirectly return to the user because the movable guide is linked in position/force with the driven object;
- contact between objects, which remains classic until the constraint is being activated.

Until the mechanical constraints are being activated, and then the collision de-activated, the geometric guides are subjected to contact force reactions. At the time of the contact, the geometric guide linked to the driven object is forced to move along the fixed geometric guide. Depending on the geometry of this virtual wall, it is then possible to lead the user following preferential positions.

Figure 4 illustrate the use of geometrical guidance to position a peg over a hole. The dotted line is the geometric guide linked to the peg's axis and the centerline represent the geometric guide fixed upon the hole's axis. Initially (left picture), the peg is haptically driven by the user toward the hole. Once the two guides contact on a local point (second picture), the combination of reaction force and user's movements adjusts the peg's guide toward the fixed one, until they collide on a second point (third picture). Once the two axes concurring, the constraint is activated and the user just have to give a downwards movement to the peg, while preserving the contact between the geometric guides (right figure).

3.3 Activation of the guide and deactivation of the collision detection

The main purpose of geometrical guidance is the setting in position (and orientation) of the driven object, in a

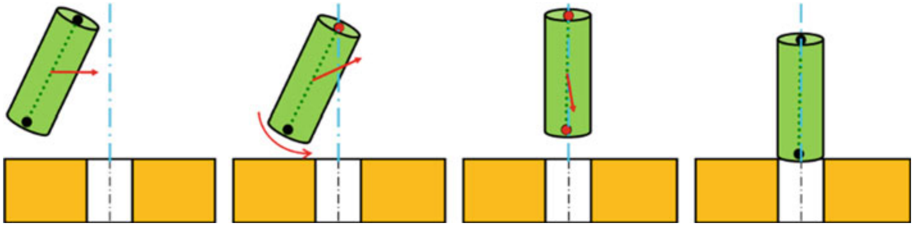

Fig. 4 Illustration of geometrical guidance, for a peg-in-hole insertion in 2D

configuration where it is gatherable with the fixed object. Once the objects are in position, the application disable the collision detection between those two objects, and activate a kinematic constraint that correspond to the assembly task. In the peg-in-hole example, this constraint is created and applied when the peg is just on top of the hole, i.e., when their respective axes are coincident. The kinematic constraint, which has been set at the early beginning of the simulation, is then applied between the driven object and the fixed objects, without interruption of the real time simulation. The apply of kinematic constraint leads to a reduction of the number of degree of freedom.

The transition between the free mode of control and the assisted mode of control, corresponds to the transition between dynamic/quasi-static and kinematic physics. To switch between those modes, one solution is to insure the continuity of the object's velocity between the time of collision de-activation and the time of constraint activation. The perception of this change must be transparent for the user and must ensure the continuity of its movement, forbidding efforts instabilities.

4 Application

A complete application of our method is presented here: the insertion of one peg (cylinder on the Fig. 5) in one hole of the same diameter. This kind of canonical task has been largely studied in the field of teleoperation and interactive simulation. The assembly then follows a specific scenario:

- the user control a peg with a haptic device and explore the virtual scene;
- the user inserts the peg in its corresponding hole;
- the user extracts the peg once it is well assembled (bottom of the hole).

In this example, the objects are tesselated and exported from CAD software, so that their assembly is not possible in classical VR applications that use collision detection. We avoid detection problems by applying VCG on the representations of the original objects. This assembly task obviously involves a two DOF relative motion, which is represented by a slider-pin joint. The first step is the placement of the peg over its corresponding hole (left part on the Fig. 5). This position is defined as the alignment of the axis of the peg and the axis of the hole. In order to position the axis, the geometrical guidance is applied between the axis of the peg (dotted line) and a geometric guide fixed on top of the hole (crossing plans). The second step is the insertion of the peg in the hole (right part on the Fig. 5). Once the peg is positioned, we deactivate the collision detection and simultaneously activate the slider-pin kinematic constraint. The axis of this joint is theoretically the same as the hole's one, and gets an end-stop corresponding to the depth of the insertion. Then reducing the number of available DOF, the object is forced to move along the trajectory, until it reaches the artificial bottom of the hole.

The methodology proposed in this article can be easily implemented with a 6-DOF haptic device. In order to create the haptic rendering of the artificial and natural constraints, and then implement the VCG, we have used a Haption Virtuose haptic device. The method has been implemented on a VR platform with a dynamic management of collision. Despite the gaps induced by LMD collision detection, we have proved the pertinence of our approach by succeeding in assembling CAD parts without modifying their geometry.

Figure 6 illustrate an assembling case that use our VCG. It represents a wide-size assembly of a windscreen-wiper in a car chassis. This industrial case involves CAD objects that have been exported in STL format. The assembly of the windscreen-wiper engine is defined as a couple of slider-pin insertions. Then it uses two simultaneous guidances to adjust the engine in the car frame. As an extension of the peg-in-hole application, we add two couples of geometric guides to the scene and activates two kinematic constraints once the engine is positioned.

5 Experiments and results

In this section we propose to study the performance and usability of our VCG method. We perform an evaluation of the guidance method, for the activation of kinematic constraints in a peg-in-hole assembly. The purpose of these experiments is to evaluate the usability of our method. In our experience, the user performs enchained tasks of assembling and disassembling, to verify that the user can perform tasks with our guiding method (effectiveness). We then

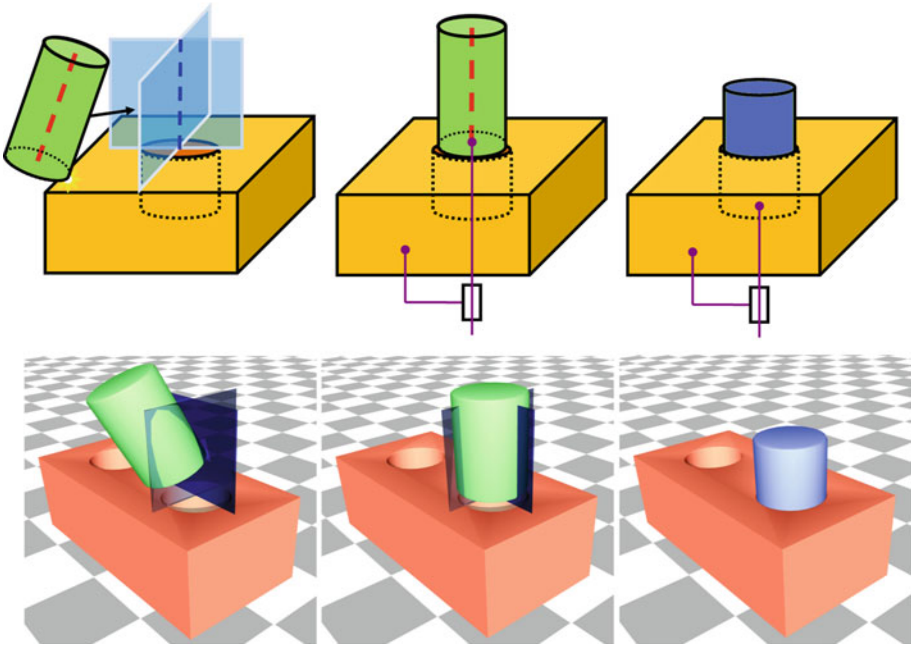

Fig. 5 Different steps of the peg-in-hole insertion using VCG

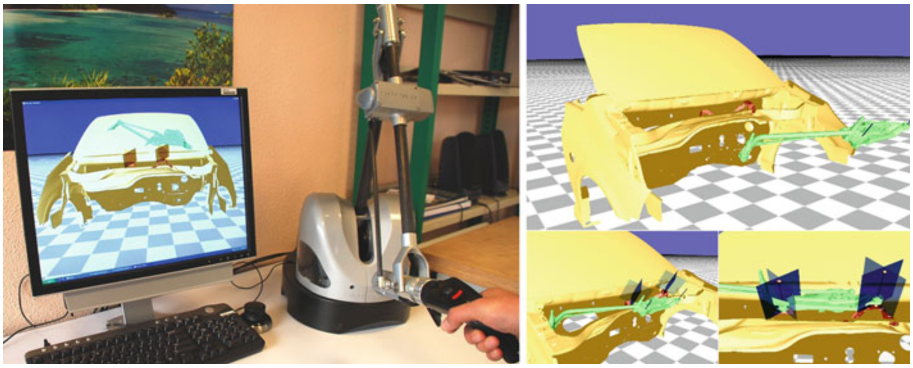

Fig. 6 Assembly of a windscreen wiper in a car frame using a Virtuose haptic device

evaluate the contribution of our method on the user performance (efficiency).

For our experiments, we use a haptic device Haption Virtuose 6D35-45. This device has its own computing unit, which interacts directly with our application while respecting the real-time constraint real-time constraint. This device proposes a six axes force feedback with a large workload, especially suitable for handling operations of virtual objects in scale 1. We perform our tests on a laptop that has a Centrino processor 2.33 Hz, with 2 GB of RAM and an NVidia Quadro FX2500. For our experiment, we use a physics engine called XDE, for eXtended Dynamic Engine, developed by CEA-List. This engine manages the behaviour of complex rigid solid following a non smooth dynamics scheme (time-stepping method).

The VCG applies a kinematic constraint between two objects, on which the collisions have been locally disabled. For simple tasks such as insertions, this constraint assists the finishing of the assembly. We make the assumption that the degrees of freedom of a inserting peg, are those of a sliding-hinge joint, if we ignore the friction. This joint is then activated when the peg can enter in the hole, i.e., when its axis is coincident with the axis of the hole. Before the launching of the simulation, we manually define the type of kinematical constraint for each subassembly, a sliding-hinge one, and we manually introduce the geometric guides to associate with. The geometric guidance we use involves a moving guide associated to the axis of the peg, and fixed guide associated to the hole (see Fig. 5).

We base the experience on the works of Amirabdollahian [13] who proposes a anthropocentric study on the trajectories of an inserting peg in two neighbour holes. We then simulate sequences of insertion between 4 holes (see Fig. 7). For this manipulation, the 3D objects are firstly designed on the Dassault CATIA CAD modeller and then exported to STL (triangular mesh). They are finally converted to OBJ format (polygon mesh) by Autodesk 3DSMAX and introduced on our platform.

The user controls the peg thanks to the haptic device, and inserts it in the 4 holes according to predetermined sequences.

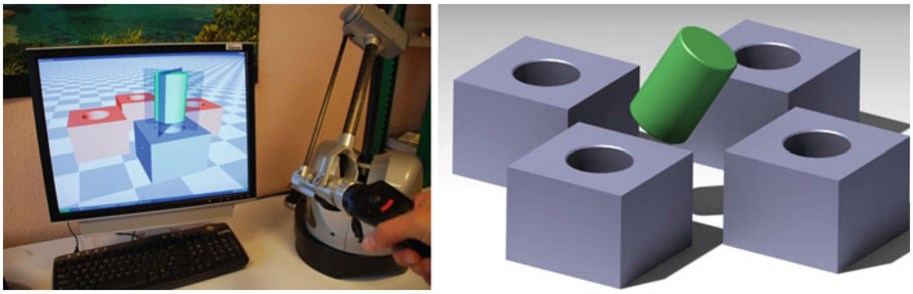

Fig. 7 Experience: consecutive insertions of the peg in four different holes

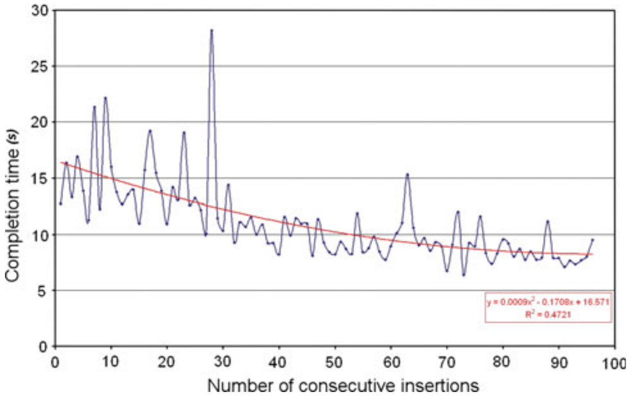

Fig. 8 Average time completion through tasks, for all the 15 users

The measurements are performed when the peg is near to its destination, then isolating the travel time between each 4 zones. Two different types of guide are tested. As a first referential case, no geometric guide is applied. In a second case, the geometric guidance, describe in preview sections, is applied.

During manipulation, those 2 cases are presented in the simulation with 6 other cases. The other cases (not presented here) vary the visualization and the contact of the geometric guides. Each case was repeated 12 times, spread over the 4 blocks leading to 96 couples of insertion/de-insertion. For each insertion task, the simulation only activates the VCG of the target hole and measures the time of completion of the user. This measurement is performed automatically in the simulation via a clock function (in C++), between the moment the peg enters into the assembly area, and the moment it is inserted into the bottom of the hole. The duration of one experiment is about 30 min, depending on the user's experience of the Virtuose 6D35-45. Fifteen users conducted experiments, 14 men and 1 woman, aged from 24 to 33 years. All users are familiar with 3D environments: 12 are experts in virtual reality, including two specialists in force feedback interfaces. No people have knowledge in virtual fixtures or interactive assembly.

5.1 Results

The statistical analysis that we describe below are based on the results of $15 \times 96 = 1440$ measures.

The 15 subjects earn significantly in speed over trials. Figure 8 shows that the average time of insertion drops over the series, regardless of the kind of guidance. For all subjects and all the 96 trials, the average time decreases from 14.53 s in the early manipulations to 8.75 s at the end. The standard deviation measures also decreased from 9.94 to 4.41 s for the entire population. This fact can be easily explained by a learning effect for about 40–50 tests, and then a stable performance value over time.

Figure 9 summarizes all performance tests. It shows the average times of insertion, for all subjects, depending on the case assembly. The average time for all subjects and all trials is 9.68 s with a standard deviation of 5.4 s. The reference case (number 1), where the user must place the peg without any assistance (either visual or haptic) proposes the worst results. The average associated time is 10.64 s versus 8.43 s for the second case. This difference of 20.8 % shows that the use of geometric guidance provide a real gain in completion time. The deviations are, for this comparison, egal to 5.35 s for the first case and 5.46 s for the second case. Those last results reflect a greater variability of behavior, meaning that the decrease of performance can mask dispersed results.

The study of these histograms allows us to highlight the results of the second case, which offer haptic interaction between the guides. The concentration measurements lower than 9.7 s (total average) is close to 60 % for case number 2, against 47 % for case number 1. The mean differences show that there is little difference between subjects, though there are real differences between the two cases. To be more precise, the level of expertise (novice, intermediate or expert) greatly influences the performances. We bet that these results are linked to a very specific task, the peg-in-hole, which is repeated many times during the same manipulation. If the learning effect and the initial level of knowledge can not be neglected, we must not extrapolate the general use of our virtual guidance system without testing it on a industrial case.

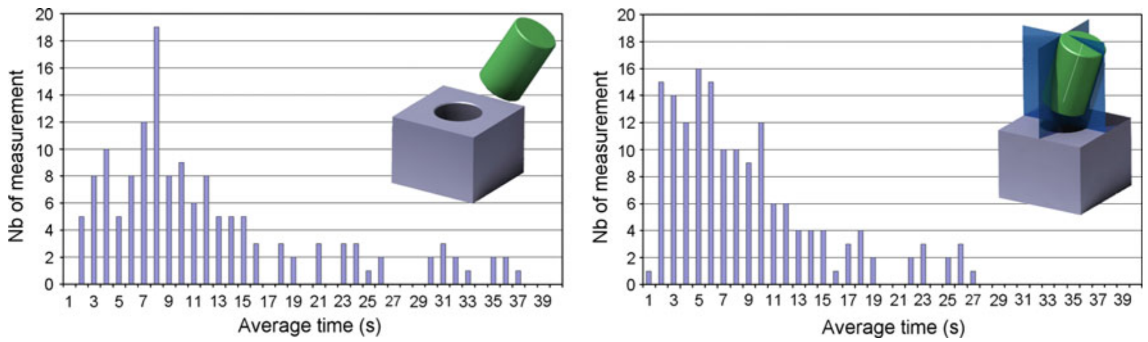

Fig. 9 Repartition of time completion (average) in function of the number of task

6 Conclusion and future work

In this article, we have proposed a new method of assembling CAD object in VR application. The complementary use of virtual fixtures and mechanical constraints in the interactive simulation leads to efficient and intuitive assembly of objects. The main advantage of this VCG method consists in working on geometries that are directly re-used from CAD application, and allow their assembly by getting round the problem of collision detection. By adding geometric guides to the simulation and making them interact with abstractions of the controlled objects, we have presented a new positioning technique. The on-line activation of kinematic constraints, authorised by the coincidence of objects geometries, assist users to perform insertion tasks. Those artificial constraints, which are determined by the force cues and constrained motions, are applied to guide users through a desired path. In addition, the case study of this paper demonstrates the feasibility of use of virtual constraint guidance for assisting the user in performing an assembly task. The first result on a peg-in-hole insertion shows how useful and intuitive the method is, providing hands-on experience to the user and a better understanding of the task. Moreover, we have investigated the efficiency of our method with respect to the task realization duration. The study was a the rigorous evaluation of the VCG for human–computer interaction processes in assembly. This user-centred study, based on a peg-in-hole application, allowed us to evaluate the usability of the guidance, and to analyse the performance of task completion. Even though the VCG provide efficient results for insertion tasks that use combination of slider-pin joints, we would like to implement our method in complex assemblies. To generalize our approach, we will have to describe more precisely the possible actions of assembling, then modelize them with adequate constraint and geometric guides. In addition, we will have to automatize the generation of the virtual guides thanks to the contact identification between objects as proposed by [14].

Acknowledgments This work was supported in part by the grant 06 TLOG 031 of ANR, in the context of the RNTL—Part@ge project.

References

1. Dewar, R., Carpenter, I., Ritchie, J., Simmons, J.: Assembly planning in a virtual environment. In: Conference on Management and Technology, pp. 664–667 (1997)
2. Yuan, X.: Interactive assembly planning in virtual environments. In: IEEE International Conference on Intelligent Robots and Systems, vol. 2, pp. 1462–1467 (2002)
3. Rosenberg, L. B.: Virtual Textures: perceptual tools for telerobotic manipulation. In: Virtual Reality Annual International Symposium, pp. 76–82 (1993)
4. Tching, L., Dumont, G.: Interactive simulation based on non-smooth contact dynamics: application to haptic rigid-body simulations. In: Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology, pp. 87–90 (1998)
5. Kockara, S., Halic, T., Igbal, K., Bayrak, C., Rowe, R.: Collision detection: a survey. In: IEEE International Conference on Systems Man and Cybernetics, pp. 4046–4051 (2007)
6. Jimenez, P., Thomas, F., Torras, C.: 3D collision detection: a survey. Comput. Graph. **25**, 269–285 (2001)
7. Gottschalk, M., Lin, M. C., Manocha, D.: BBTree: a hierarchical structure for rapid interference detection. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH 96, ACM Press, pp. 171–180 (1996)
8. Johnson, D., Willemsen, P.: Six degree-of-freedom haptic rendering of complex polygonal models. In: Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, p. 229 (2003)
9. Jayaram, S., Wang, Y., Jayaram, U., Tirumali, H., Lyons, K., Hart, P.: VADE: a virtual assembly design environment. In: IEEE Computer Graphics and Applications, no. 6, pp. 44–50 (1999)
10. Bourdot, P., Convard, T., Picon, F., Ammi, M., Touraine, D., Vzien, J. M.: VRCAD integration: multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models. Computer-Aided Design (2008)
11. Gupta, R., Whitney, D., Zeltzer, D.: Prototyping and design for assembly analysis using multimodal virtual environments. Computer Aided Design, Special issue on VR in CAD, pp. 585–597 (1997)
12. Seth, A., Vance, J., Oliver, J.: Combining geometric constraints with physics modeling for virtual assembly using SHARP. In: ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (2007)
13. Amirabdollahian, F., Gomes, G., Johnson, G.: The peg-in-hole: a VR-based haptic assessment for quantifying upper limb performance and skills. In: IEEE International Conference on Rehabilitation Robotics, pp. 422–425 (2005)
14. Iacob, R., Mitrouchev, P., Leon, J.: Contact identification for assembly-disassembly simulation with a haptic device. Vis. Comput. **24**(11), 973–979 (2008)