A framework to design 3D interaction assistance in constraints-based virtual environments

Mouna Essabbah, Guillaume Bouyer, Samir Otmane, Malik Mallem

HAL Id: hal-01020152
https://hal.science/hal-01020152

Submitted on 23 Apr 2021

A framework to design 3D interaction assistance in constraints-based virtual environments

Mouna Essabbah · Guillaume Bouyer · Samir Otmane · Malik Mallem

Abstract The equilibrium of complex systems often depends on a set of constraints. Thus, credible virtual reality modeling of these systems must respect these constraints, in particular for 3D interactions. In this paper we propose a generic framework for designing assistance to 3DI in constraints-based VE that associates constraints, interaction tasks and assistance tools, such as Virtual Fixtures (VF). This framework is applied to design assistance tools for molecular biology analysis. Evaluation shows that VF designed using our framework improve effectiveness of the manipulation task.

Keywords Virtual reality · 3D interaction · Framework · Complex environments · Constraints · Assistance model · Virtual fixtures

1 Introduction

Complex domains or systems such as scientific data exploration or robot tele-manipulation are ruled by a set of constraints, for example on the tasks to be performed (e.g. time, safety, etc.) or on the behaviour of dynamic elements (e.g. physical or chemical laws, etc.). These constraints are identified, studied and modeled by the experts of each domain. Virtual Reality (VR) simulations of these systems have also to integrate these constraints in order to be credible. This integration should be at every level of the application: 3D modeling, 3D user interaction (3DI), behaviour of the virtual environment (VE), etc. This work is part of a larger project,

M. Essabbah
French National Research Agency
E-mail: mouna.essabbah@agencerecherche.fr

G. Bouyer, S. Otmane, M. Mallem
IBISC, Univ. of Evry-Val-d'Essonne
E-mail: FirstName.LastName@ibisc.fr

which aims to provide an application for 3D interactive modeling of chromosomes. After an automatic generation of chromosomal 3D models from physico-chemical rules, the biologist can manually edit the 3D model to improve/correct it. However, the problem is that he/she may alter the physico-chemical reality of the 3D model. The question is then how to help the human user to maintain the initial constraints during the interaction phase.

Therefore, the purpose of our work is to provide assistance to 3DI in constraints-based VE, not only to improve user's performance (e.g. precision or speed), but also to improve the credibility of his/her experience by ensuring the coherence between the building rules of the VE and the interactions with it. This issue arises in a variety of application domains and it is suggested that this research addresses the more general issue in domains other than biology. Therefore, we propose a new framework for designing assistance tools from the specification of the task to achieve and the constraints imposed by the application domain. The conceptualization we suggest provides a useful guideline. The contribution of this research is: 1) a new framework for designing assistance to 3D interaction called CTT that associates constraints, interaction tasks and assistance tools, such as Virtual Fixtures (VF), 2) a new formalism of VF (guides), and 3) a case study which demonstrates the implementation of the framework and evaluates the benefits of resulting VF with non-immersive and immersive devices.

The paper is organized as follows: section 2 presents the related work to 3DI assistance and constraints; section 3 describes our assistance model based on the relation between constraints, assistance tools and 3DI tasks; section 4 presents the application of Molecular Biology (MB) we chose to test our model; section 5 de-

tails the evaluations we have conducted and discusses the results.

2 Related work

The interaction integrity of a virtual system is its ability to conform to actions that could be done in reality. It could be equivalent to the behavioral realism. To ensure the interaction integrity in a VR system, assistance is proposed to users. We can define assistance as a set of tools that the system provides to users to help and guide them while interacting with virtual worlds. It may be in various levels from lowest to highest (Fig. 1). For example, assistance may be a simple transmission of information (e.g. written or vocal message, visual trajectory to specify a path to follow). Support can be made by other users either through a direct exchange (e.g. verbal instructions) or by active cooperation in the application (e.g. sharing of remote entities). Higher level assistance can be software, with predictive algorithms (e.g. to anticipate the impact of a task on the system) or multi-sensory indicators (i.e. audio, visual or haptic): force feedback to help users assemble entities or to simulate collisions between objects [13] [14] [19], etc. It can also be physical, through tangible interfaces [17].

In some studies, constraints are implicitly incorporated into the 3DI technique. For the navigation task, a 2D map can be used to characterize a shift from the current location to next one while being aware of the constraints that compose the virtual world [2]. A virtual deformable ray (3D curve) can represent a path, avoiding obstacles [39]. In the case of manipulation, the movement of objects can be restricted by constraints [38] so they are implicitly preserved. When the target is remote or small, specific areas can act on the effector (by attraction/repulsion) to improve the accuracy of conventional virtual hand technique [29].

In other works, constraints are explicitly represented and assistance tools are virtual fixtures (VF). The VF improve the performance of a human operator in tasks of remote manipulation with transmission delay [35], using different sensory channels (audio, visual and haptic). Also known as virtual mechanisms [7], VF can enhance both the execution time, quality and precision of tele-operation tasks [22], [23]. They are used to help in the design and the assembling of objects in CAD [30], to support tele-manipulation in Augmented Reality [28] or in assistance to collaborative 3D telework [29]. Haptic VF can also guide the user along a specified path, prevent access in restricted areas [1], or limit surgeons' movements in critical operations [34]. The use of haptic feedback was evaluated and shows

better performance than classic audiovisual interfaces [27].

All these works do not explicitly refer to "guides" and "constraints" at the same time. They are implicitly integrated in those applications, for example by modifying the interaction techniques or adding ad hoc elements in the VE. This lack of genericity leads to modify the entire system whenever there is a new constraint.

Therefore, some researches focused on the generic formalization of VF in order to ensure their reusability. VF have been first formalized for graphic assistance, limited to the visual modality [28]. They have recently been formalized in the case of haptic, based on mechanics [33]. It was applied to follow a trajectory in a virtual training application for surgeons. These two models make guides configurable and scalable: standard VF can be configured to adapt to new situations or tasks. Though, they still disregard the constraints of the system.

Some works are interested in the concept of constraints-based environment. For example, in maintenance simulation, [24] proposed a manager to support physical realism and interactive assembly and disassembly tasks within VE. However, this approach is limited to the geometric constraints and so restricted to the CAD-like applications. Otherwise, [36] used haptic based communication approach to overcome constraints related to collaborative virtual environment. Despite that, it is still limited to a particular domain. Our paper addresses guidelines for more general constraint management.

In conclusion, we observed that, on one hand, there are generic 3D interaction techniques for acting in a virtual world. On the other hand, there are assistance tools to 3D interaction such as VFs (or even ad-hoc feedbacks). What we propose is a new framework to design assistance tools adapted to the targeted VE. This framework is based on the specification of the task to achieve and the constraints imposed by the application domain to offer a tailored assistance. It explicitly integrates the constraints of the VE, to support the design of generic and multimodal VF. This formalism is based on three components: Constraints, assistance Tools and interaction Tasks (CTT).

3 The CTT assistance framework

3.1 Assistance basis

Such assistance covers complex areas where virtual representations must match reality. We are not looking for visual realism but behavioral one, that's why we are interested in the impact it can have on 3DI tasks.

The behavioral realism is determined by the constraints identified by the designer and that are required for 3D interaction to be sufficiently credible. Integration of assistance in the interaction process with VR environments aims to reduce the user cognitive load [40] on how to interact, due to physical constraints or software techniques used. For the same reasons, assistance includes constraints of the application domain directly into the interaction system, transparently for user. In other words, assistance helps the users obey constraints or at least reminds their existence.

In Figure 1, we classify different assistance types regardless of the used tool. We do not consider the technical means used for assistance but only its objective. Our goal is to understand the relationship between the required assistance level and the VF type. Thus, the VF specification will be done according to the needed assistance. For constraints-based 3DI, we observed different assistance levels: to inform of the constraints existence, to inform that user's actions are not complying with constraints, to inform about the current state of compliance or exceeding with constraints, and finally to automatically prevent exceeding constraints.
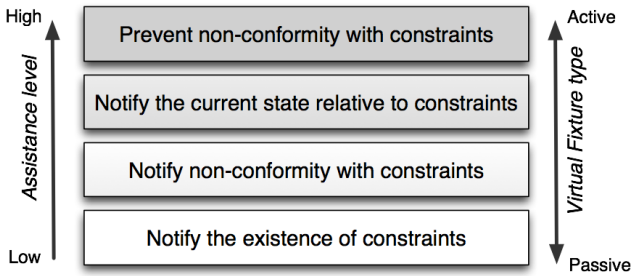

Fig. 1 Different assistance levels in constraint-based VE.

To ensure 3DI according to domain constraints, we propose the assistance model CTT (Constraint-Tool-Task). This model has three components (see Fig. 2):

- Constraint: It represents the system constraints. According to our classification, it can be local and/or global. The constraint limits the interaction task (section 3.3).
- Task: It represents the possible actions in an application. The four standard 3DI tasks are: navigation, selection, manipulation and system control (section 3.2).
- Tool: It is the interface between the other two model components. The assistance tool allows performing a task according to all rules imposed by the application domain (i.e. constraints). It can be tangible (instruction manual), human (co-user) or software (audio, visual and haptic VF) (section 3.4).

Figure 2 shows globally the relationship between the three components of our model: we identify the various considered constraints of the application domain and formalize them. Then, from this formalism we can deduce the geometric primitives that will form the VF. Then, we must attribute to VF the modalities we want to use. Finally, given the constraints, we have to specify for which 3D interaction tasks guides must be enabled.
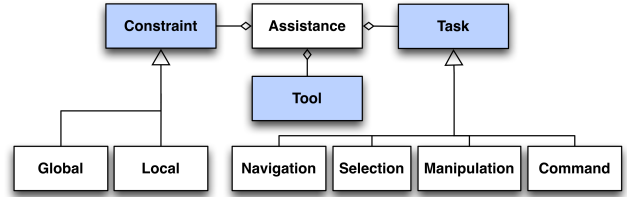

Fig. 2 Classes defining the CTT framework.

3.2 Task

3DI is an essential aspect of VR, it allows the user to act in the virtual world. Bowman defines three kinds of natural actions: navigation, selection and manipulation of objects [2]. Finally, a fourth type of task, more specific to computer science, is the system control. They are carried out through interfaces, which may be sensory or motor, supplemented by software tools and methods called interaction techniques. These 3DI techniques can be generic or specific for each task. Here are some well-known examples:

- Navigation: Moving or changing the point of view in a scene with the technique Gaze-directed steering or pointing, Grabbing the air [25], or a Map-based travel [5].
- Selection: Designation of one or more objects for a given purpose by the technique of simple virtual hand, ray-casting [20], sticky finger [31] or arm extension [32].
- Manipulation: Changing the properties of objects by the technique of simple virtual hand, HOMER [4] or Scaled-world grab [26].
- Control system: Changing the system state or interaction mode with graphical menus, voice commands and gesture interaction.

Table 1 Some issues in VR applications, related to the task, the virtual environment, the real environment or the users.

Task	
Precision and/or Safety	Spatial requirements for navigation in the scene, selection and manipulation of objects. Otherwise failure or danger. Ex.: put an object on a target, cut a virtual organ, follow a trajectory, etc.
Execution time	Maximum total time imposed to complete a task. Otherwise failure or danger. Ex.: perform a navigation towards a target, complete several maintenance tasks, etc.
Virtual Environment	
Laws	Often determined by the domain (physico-chemical, biological, mechanical, etc.). Result in geometrical and behavioral constraints: positions, movements, collisions of autonomous virtual objects, DOF of the virtual hand, etc. Corresponding VE is sometimes referred to as "constraint-based environment" ([15]): scientific simulations, interactive assembly, etc.
Design needs	Specific requirements about the aesthetics or the composition of the scene.
Massive data	Complex scenes limit calculation capacities and make it difficult to perceive and understand these information.
Real Environment	
Interfaces	Characteristics and physical limits (weight, shape, DOF, FOV, accessible physical space, etc.).
Transmission time	Related to tele-operation: the distance between the two sites creates a delay between ordering and task activation.
Environmental conditions	Light, sound conditions, etc. during the execution of the application.
Users	
Physical and Cognitive resources	Handicap may limit the sensori-motor capacities of the user. The amount of available cognitive load also varies depending on his/her profile and the task performed.
Collaborative work	Requires interaction management for coordination (rights, priorities, intervention order, etc.) and specific clues (in particular for the awareness of presence).

3.3 Constraints

In VR, the concept of constraints can be viewed as the opposite of affordance[1] because it limits the potential actions of an object. The word "constraint" itself has a negative connotation because it refers to something that limits us, but constraints are necessary, both in real and virtual situations [3]. This concept has been applied in the field of real time graphical simulation, interaction or haptic rendering: "constraints" are spatial or semantic relationships between virtual objects that have to be respected. These relationships constrain the position, the kinematic or the behaviour of the objects. Constraints can be resolved using either physical equation systems that are computationally intensive [18], dedicated hardware [9] or software architectures (e.g. Geometric Constraint Manager [24], Constraint Logic Programming (CLP) [8]). Smelik et al introduced a new use of the word with "semantic constraint": a control mechanism imposed on the procedural generation of virtual environments in order to satisfy explicit designers intent over a specific area (e.g. a line of sight between two positions). It is composed of "feature sub-

constraints" mapped to low-level operations (e.g. the height of trees) [37].

Indeed, constraints are not always geometric or even virtual: table 1 shows a list of possible issues, situations or requirements which can have a constraining influence on VR interactions. Some are inherent in a domain, due to the simulated environment or required specifically by the task, others are imposed by the real environment or the devices. We can then distinguish "positive constraints", that users should respect to conduct a meaningful activity, and for which they need assistance from the system; and "negative constraints", that designers and users have to deal with.

Our proposed framework, which maps specified constraints with formalized virtual fixtures, can apply to these types of constraints.

3.3.1 Proposed classification

Constraints imposed by the environment or the task affect both virtual entities (e.g. spatial form, position, etc.) and their behaviour (e.g. possible movement, collisions, etc.). However, we also identified constraints affecting the whole system.

Contrarily to the classification proposed by [3] that identifies the constraints according to each 3DI task independently from the rest of the system, we are inter-

---

[1] From the verb "to afford": the ability of an object to suggest his own use [16]

ested in the behaviour of one or more objects in their environment. For instance, object properties change while it is manipulated (e.g. position, orientation, etc.) and this can influence surrounding objects. Therefore, we study the impact of constraints not only on each task but also on the whole 3DI system. In parallel, we also take into account the consequences of constraints on an object and on all the virtual entities.

We propose the following classification to identify the different constraints types:

Global: This is the main type of constraint. It concerns the final result or outcome of the application. A global constraint is related to:
  – the virtual objects set, such as an overall bounding volume or their relative positions.
  – the whole 3DI system, such as accuracy or speed expected for the task.
Local: This type represents a sub-constraint of global type that is associated to a task (or an application function). A local constraint may concern:
  – a 3D object (entity of the VE), for example by imposing a geometric shape.
  – 3DI task, such as a confined manipulation space, or a limited distance for navigation.

Figure 3 represents a VR system with its specific constraints and assistance toolbox. The VE main modules are the set of virtual objects and 3DI system. The local type of constraint refers to elements of each of the previous subsystems. This type concerns 3D objects and 3DI tasks. Thus, if we consider that a constraint is a parameter of our system, so there may be sub-constraints in our application. The constraints categorization contributes to the specification of the corresponding VF. Indeed, this parameter indicates when the VF would be enabled. For example, if a VF is associated to a local constraint such as precision in selection, it will be activated only for the selection task. On the other hand, if the VF is related to a global constraint, it should be active all the time.

3.3.2 Constraint specification

The constraint specification is based on observation we have done of different constraints in various application domains (examples in table 1). A complex environment is often organized according to several correlated constraints. Therefore, we need to isolate each constraint separately and then determine the process that connects all of them. Thus, as in the rule-based systems, constraints can be considered as unit rules whose relationship is managed by logical operators (AND, OR, XOR, etc.). The first stage of the design is precisely to decompose the constraints into logical rules.
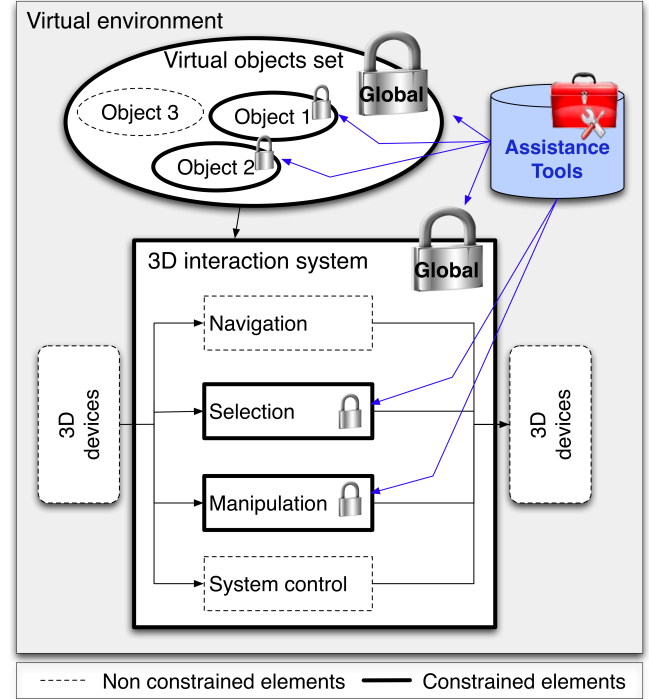

Fig. 3 Example of contraints distribution (represented by padlocks) in a VR system.

Table 2 Some examples of constraints specification.

Constraint	Specification
Constraints between objects: O_1 and O_2 must not collide	C_1 and C_2 are respective centres of O_1 and O_2, d the distance between them, thus: $\forall\ O_1$ and O_2 position, $d < d_{m}ax$
Destination/ path: O should follow a circular path	$\forall\ O$ position, its coordinates must follow the equation that describes the trajectory imposed $+-\ \epsilon$
Temperature: The VE temperature T should not exceed T_{max}	$\forall\ T$ at any time, $T < T_{max}$

Thus, a constraint can be represented by a quantifiable parameter C_i that will be compared with one or more thresholds. Depending on the value of this parameter, an associated VF is activated. Table 2 provides some simplified examples of constraints specification. The goal of this specification is to implement each constraint by translating the rule in simple calculations, tests and parameters extraction, in order to execute the rendering functions of their associated VF.

3.4 Tools

The *tool* component of our assistance model results from the specification of the *task* to achieve and the imposed *constraint*. The assistance tool is at the heart of the assistance model. We choose to use virtual fixtures as assistance tools.

3.4.1 Virtual Fixtures formalism

Our specification is based on a formalism proposed by [28]. The Otmane's formalism is built on a data structure composed of fields defining the properties of a VF (Fig. 4):

- VF_ID: Allows the identification of a VF
- VF_Type: Defines the VF type (simple, composed, passive, active). If the guide is "composed" then it contains links to other VF (e.g. using a chained list). In the case of "active" guide, it informs us if it is "repulsive" or "attractive". This field is highly dependent on the desired assistance level (Fig. 1): if assistance is low, the VF will be passive, but if it is high, the guide will be active.
- Referential: Contains information about position and orientation (3D coordinates) of the guide in the VE.
- Attachment: Specifies whether the guide is "static" (fixed in one place or attached to a particular object) or "dynamic" (appears after an event, for example, when the avatar enters the precise manipulation area of a virtual object). In the first case, it contains the name of the virtual object or the point at which it is attached (otherwise *NULL*).
- Influence_Area: Contains the analytical equation (static or parameterized) defining (partially or completely) the guide shape.

As described above, this formalism is limited to the specification of geometric VF with only a visual rendering. We expanded it to adapt the assistance in constraints-based 3DI and to encompass a broader category of VF that may have a multimodal rendering (audio and haptic). Moreover, we group the geometric primitive and the guides function for a clearer specification. In particular, it is essential that VF include a reference to the constraint that it represents and to the task for which it is activated. The data structure that represents our VF formalism (Fig. 4) must take into account three new components:

- Constraint: Contains settings related to domain constraints, such as a geometric equation defining a manipulation space. Assuming that each constraint

can be modeled by a relational equation (Constraint_Equation), the guide should inform this relationship, but also its parameters.
- 3DI_Task: This field identifies to which 3DI task this guide is related. This can affect the activation function of a guide (e.g. VF for selection only).
- Modality: In order to be perceived by the user, the VF must provide a visual, auditory or haptic rendering. Thus, used tool must include information on the type of sensory modalities that are operated.
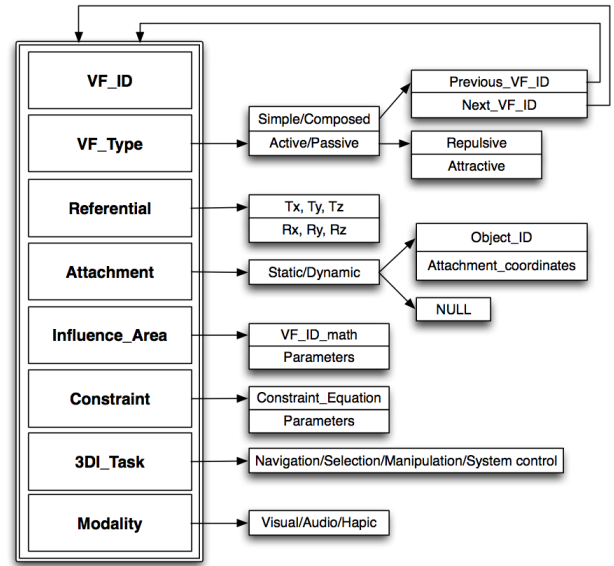

Fig. 4 VF data structure for assistance in constraints-based 3DI.

For example, the temperature of the VE is a global constraint that can be represented by a planar VF orthogonal to the view of the virtual camera. Depending on temperatures threshold(s), the guide may change color or opacity. For instance, higher is the temperature, redder is the scene. If another constraint is to not entering a perimeter where the temperature is above T_{max}, the guide can use haptic feedback to be repulsive. Here is an example of a VF specification:

- VF_ID: temperature_guide;
- VF_Type: simple, active;
- Referential: root of the VE;
- Attachment: static;
- Influence_Area: a plan;
- Constraint: $\forall\ T,\ T < T_{max}$ (see table 2);
- 3DI_Task: manipulation; Modality: visual, haptic.

3.4.2 VF operating cycle

The operating cycle of a VF is given in Fig. 5. In general, the activation of a VF is an event triggered by a
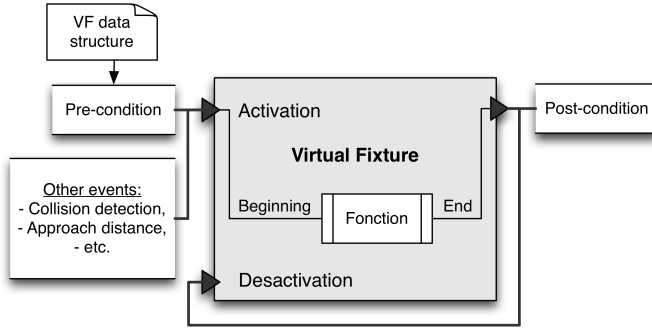
Fig. 5 VF operating cycle.

membership test of a point to a VF (e.g. position of the effector). This is a "Pre-condition" (or activation condition). In this case, the guide properties are recovered from the data structure relative to this guide to define this pre-condition. It is also possible to activate a guide by another event such as collision detection or approach distance.

When the guide is enabled, we have to determine actions or operations to be performed: the "function" of the guide. Generally this function comes from a combination of actions realized by user and renderings defining the type of the guide. The function can also be limited to the execution of an operation defined by the guide (e.g. allows the automatic execution of a task).

The guide function has no effect on the virtual world and the operator, when the application has reached a desired end state, or the end of the automatic execution of a task. This disables the guide and defines the "Post-condition".

3.5 Model design

Now we are interested in the concrete implementation of the assistance model without considering the rest of the system (because the application architecture may differ).

We start from the simplified class diagram of figure 2 and we focus on the *Tool* component of the CTT model. Figure 6 illustrates the new class diagram involving major classes that define the various properties previously defined. We propose an architecture for the CTT model based on different structures and different levels of abstractions to reflect the characteristics of some VF. The distribution of characteristics on several classes allows to distinguish on one side the general appearance and on the other side components and specific functions of each VF.

Besides the global attributes (VF_id, VF_Type, $referential$, etc.), a VF is composed of primitive and sensory modalities. The *Primitive* class represents the

geometry of the guide (i.e. its influence area). It is possible that different guides have the same primitive, which justifies the composition relationship between $VirtualFixture$ and $Primitive$ class.

Modalities employed by the VF are represented by the $Modality$ class whose subclasses are three possible sensory channels (visual, auditory and haptic). This class represents the perceptible aspect of a guide, its concrete rendering. The aggregation relationship between $VirtualFixture$ and $Modality$ indicates that a guide can be expressed in one or several modalities and a modality can be used by zero or many guides. Then, the $Constraint$ class is related to the $VirtualFixture$ class since a guide implements a constraint and a constraint may be associated to several guides. On the other hand, a VF is active during one or more tasks. In contrast, during a task, there could be zero or several guides. VR application may be subject to various constraints related to its associated real environment. Thus, to define the appropriate VF, the first step is to model each constraint by an equation defining $Constraint$ class. Then the choice of parameters determines the state of constraint and its influence degree on the system. In addition, VF must incorporate the type of tasks ($Task$) during which they will be active, and the type of modalities ($Modality$) they will operate.

4 Case of study: in virtuo project

4.1 Context

One of the aims of Molecular Biology (MB) is the study of molecules' 3D structure. *In silico* experiments (i.e. computing simulations) for 3D modeling usually use automatic approaches. However, these approaches have limits: time-comsuming computation, non-modifiable generated model, false-positive results, etc. The interactive contribution of expert knowledge, during the automatic modeling process, could overcome some limits of the usual computational methods. It involves placing the biologist in the centre of virtual experiments, rather than an observer of automatic simulation results. This is what we call hybrid approach [11], that combines the advantages of *in silico* experiments, Human-Computer Interaction (HCI) and VR: natural interaction, immersion in the VE, multimodality, etc. The result of this approach is the creation of *in virtuo* experiments which have three components: the 3D modeling, the visualization and the 3DI.

We think CTT can be part of the solution to reach this hybrid approach. Indeed, it formalizes the close links between the 3DI tasks, the constraints which are
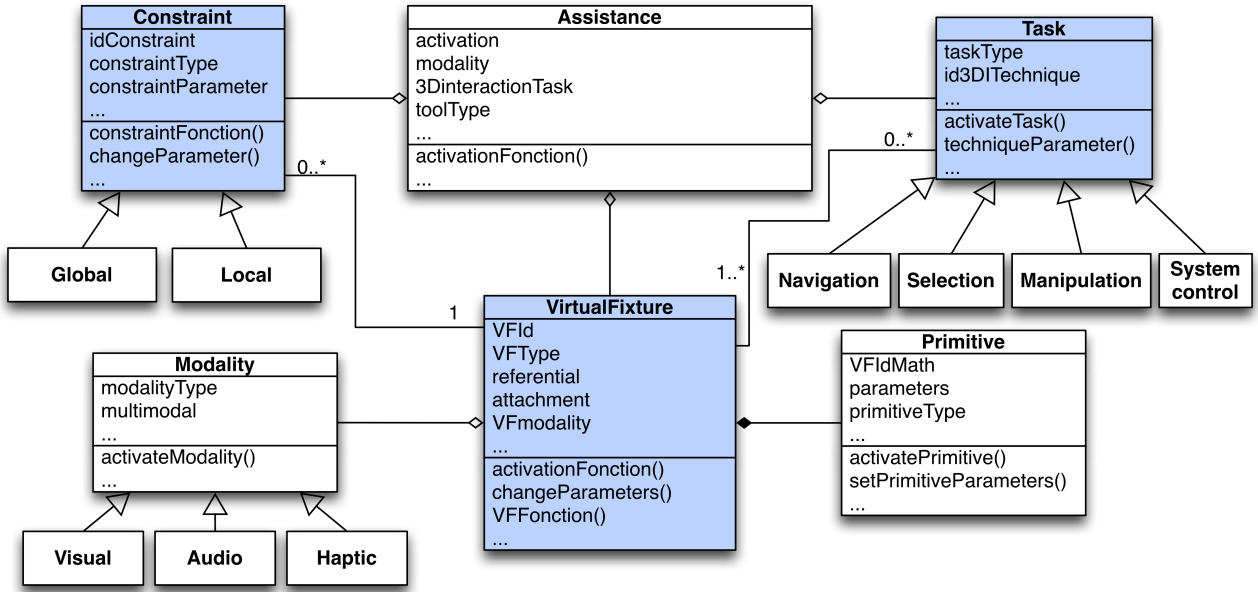
Fig. 6 Set of classes containing attributes and methods that carry the VF functionality.

the parameters of the modeling algorithms, and the assistance tools which ensure these constraints stay consistent during the 3DI.

We applied this concept of *in virtuo* Human-Molecule Interaction to chromosome 3D modeling. The architectural (i.e. physico-chemical data) and functional (i.e. biological models) constraints dictate the chromosome spatial organization. Structures of chromosomes built with this hybrid system respect these constraints. They could be considered more credible, on one hand compared to the results of pure algorithmic methods, and on the other hand compared to knowledge-based manual building.

4.2 Application of the assistance framework

Our CTT assistance framework has been applied to the *in virtuo* project, giving the following elements.

4.2.1 Constraints

Architectural constraints of the 3D model (Fig. 7) based on physico-chemical data:

– An angular constraint, due to the curvature energy of the molecule,
– A volumic constraint, due to limited volume in which the molecule evolves (chromosome territories),
– A self-avoidance constraint, resulting from the diameter and the persistence length of the molecule.
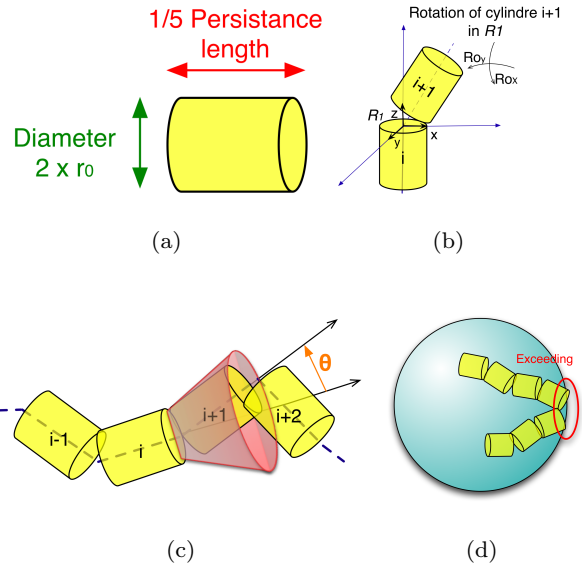

Fig. 7 Constraints in the analysis of a chromosome.

4.2.2 Tasks

The application integrity must be preserved during the different 3DI tasks: navigation, selection, manipulation and system control. However, the assistance deals particularly with the manipulation task. We propose a low assistance to selection, but a mean assistance to manipulation since it is during this task that the 3D model may lose its physico-chemical integrity. Navigation is not the most commonly used task in this application,

although it is possible for a visual exploration. Moreover, neither navigation nor system control are subject to constraints, in this case.

4.2.3 Tools

Assistance multimodal tools vary from passive informative virtual fixtures (VF) to restrive ones. Each VF provides assistance to a 3DI task using one or more sensory modalities.

Selection The designated cylinder is colored and its ID is displayed on the screen. After the validation of the selection, another color is attributed to the selected cylinder and the manipulation VF *angle_guide* is shown.

Manipulation In addition to textual information, each constraint is associated to a VF in accordance with our CTT model (their structure is detailed in table 3):

- Angular constraint: When editing the 3D model, it imposes a limit angle at each joint (between two successive cylinders). We use the *angle_guide* VF which properties are in table 3. If the angle applied during manipulation is accepted, the guide does not change. However if there is an angle exceeding, the guide turns red, a beep is activated and a visual decoupling is applied. The visual decoupling is a modality used to dissociate the user's action from the visual rendering. That is to say that when the user reaches the limit before exceeding a constraint, he/she remains free to move but the 3D model remains unchanged. The modification is blocked. In the following, we will use the term "angular constraint-blocked movement".
- Volumic Constraint: During manipulation, the *volume_guide* tests if the volume of the chromosome 3D model (or the diameter of its bounding box) is inferior to its own volume (or diameter). If the volume constraint is verified, the guide remains unchanged. However, once there is a volume exceeding, the guide takes a red color in addition to displaying a text message informing of the excess. A visual decorrelation and an auditory beep are also activated. In the following, we will use the term "volumic constraint-blocked movement".
- Self-avoidance constraint: The cylinders must be impenetrable. To check this constraint we have used the *avoidance_guide* guide. This guide uses the visual modality. Although it is unseen by users, it ensures a decoupling display when the user causes a collision between any two cylinders. The transformation caused by the user is not applied to the model if it does not respect this constraint.

5 Experiments and evaluations

Through this experimental study, we have not fully evaluated our CTT framework but tried to evaluate the provided assistance. The main question is whether the assistance tools designed using our CTT framework provide benefits to users when interacting with constraints-based virtual environments. The experiments allowed us to:

- Compare performance of manipulation of a constraints-based molecular model with and without assistance, i.e. with and without VF;
- Compare the contribution of assistance in two different interactive platforms;
- Observe users' reaction and have a first global appreciation of the multimodal assistance tools we designed.

5.1 Experimental process

We conducted these evaluations in two types of platforms, distinct in terms of level of visual immersion[2] and interaction devices. In what follows, to simplify, we use the terms "non-immersive" and "semi-immersive" to refer to these experiments.
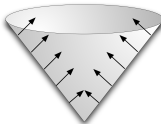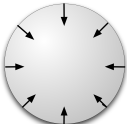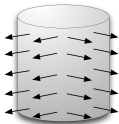
5.1.1 Non-immersive evaluations

Hardware devices are those of a personal desktop computer: a 20-inch screen for viewing, interaction with a mouse (2D pointing) for the selection and with a keyboard for manipulation (directional arrows to change the orientation of the selected cylinder). 11 volonteers participated: 8 men, 3 women. Their expertise level in HCI was self-evaluated: 5 experts, 2 intermediates and 4 novices. The evaluation included 6 tests per participant, alternately 3 with and 3 without assistance, i.e. a total of 66 tests.

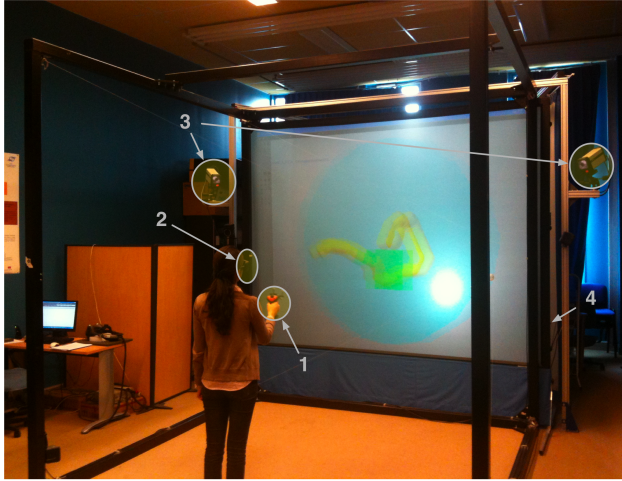### 5.1.2 Semi-immersive evaluations

Stereoscopic visualization is provided by a wide screen and stereoscopic glasses, 3DI is provided by VR devices and techniques: a hand-held flystick, ray casting for selection and simple virtual hand for manipulation

---

[2] Visual immersion can be defined as the objective level of fidelity provided by the system [6]. It depends on several factors: field of view (FOV), field of regard (FOR), display size, display resolution, stereoscopy, head-tracking, etc. Kalawsky have also adopted a techno-centered classification of immersion depending on the type of visual display and field-of-view: Non-Immersive, Semi-Immersive or Fully-Immersive Virtual Environments [21].

Table 3 Formalism of three VF designed in the *in virtuo* project.

Properties	Virtual fixtures		
VF_ID	*angle_guide*	*volume_guide*	*avoidance_guide*
VF_Type	simple, active (repulsive to inside of the guide)	simple, active (repulsive to inside of the guide)	simple, active (repulsive)
Referential	selected cylinder	geometric centre of the chromosome 3D model	cylinder
Attachment	dynamic	static	static
Influence_Area (primitive)	cone of equation: $x^2 + y^2 = z^2.(\tan \alpha)^2$	sphere of equation: $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$	cylinder of equation: $x^2 + y^2 = r^2$
Constraint	angular: if θ_1 and θ_2 are modified angles of a cylinder then $\theta_1 \in [-\alpha, \alpha]$ and $\theta_2 \in [-\alpha, \alpha]$	volume: if d_{model} is the current diameter of the chromosome bounding volume et d_{volume} is the guide diameter then $d_{model} < d_{volume}$	impenetrable cylinders
3DI_Task	manipulation	manipulation	manipulation
Modality	visual, auditory	visual, auditory	visual
Geometric representation			

(Fig. 8). 12 volonteers participated: 8 men, 4 women (4 experts, 4 intermediates and 4 novices in HCI). The evaluation included 10 tests per participant, alternately 5 with and 5 without assistance, i.e. a total of 120 tests.
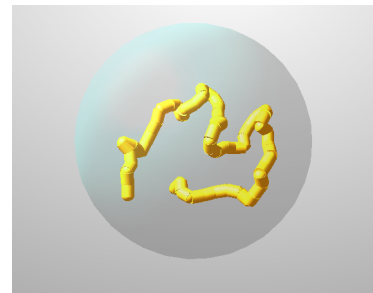

Fig. 8 Our semi-immersive plateform: (1) Flystick (2) Stereoscopic glasses (3) Infrared Cameras (4) Video projector.

5.2 Evaluation scenario

The VE is composed of a simulated 3D model of chromosome and its bounding volume. The 3D model is specially built for evaluation purposes. It represents a scale-down model (i.e. chromosome segment) generated by our geometric calculations engine [12], that can provide a substantial database of 3D models (even whole chromosome models).

The model consists of a chain of articulated yellow cylinders, connected to each other by joining spheres (Fig. 9). A blue sphere represents the bounding volume of chromosome territory. The 3D model is hierarchical

Fig. 9 Experimental VE consisting of bounding volume (in blue) and chromosome model (in yellow).

which means that the manipulation of a cylinder i will move the following cylinders ($i+1$ to n). Previous cylinders remain static.

We model the physico-chemical constraints of our application by the previously described VF (*angle_guide* and *volume_guide*).

The experiment consists of modifying the chromosome model to bring it to a desired conformation, as the expert would do in an hybrid approach. For this purpose, the participant must reproduce a form, starting from an initial conformation. To accomplish this, the participant can select and manipulate all cylinders. The task is divided in three steps where a specific cylinder must be placed in a specific area, generally by manipulating another cylinder in the chain. A cube represents an area and the target cylinder-area is indicated by a green color. The initial conditions are the same for each test: 3D model and position of areas. Validation of an area depends on a threshold distance between area and the target cylinder. The areas are dynamically displayed. Once an area is validated, it will disappear and give way to the next area. Thus, a test is to repeat three times the following steps:

1. select a cylinder and manipulate it (that is to say, change its orientation taking for reference the previous cylinder). These tasks can be repeated as many times as desired.
2. put the corresponding target cylinder in the current area to validate it.

This scenario can be compared to the game Foldit [10] that aims to create 3D proteins structures from a given puzzle with rules of molecular chemistry for only limits. To test the contribution of our CTT framework, we assigned two types of tests: 1) with assistance, i.e. with the possible activation of all VF designed section 4.2.3, 2) without assistance, i.e. without displaying the constraints and without VF. Apart from that, the experiment is identical for each of these two types of tests.

Before starting the experiment and during a phase of familiarization, we explain to the participant what is required. Then he/she alternates a test with assistance and a test without assistance. The total duration of an experiment for a participant is ten minutes on average. At the end of the experiment, the participants fill out a questionnaire on qualitative feedback from the experience.

Our prediction concerning assistance is that the VF will offer lesser constraints exceeding movements especially in the semi-immersive experiments.

5.3 Result analysis

We measure at the end of each test, the following parameters:

- Total duration of a test (T)
- Number of angular constraints exceeding movements on the X axis (N_b_dX)

- Number of angular constraints exceeding movements on the Y axis (N_b_dY)
- Number of volumic constraints exceeding movements (N_b_dV)
- Number of validated areas (N_b_Z)
- Failure : the task (i.e. three successive validations) has not been completed in a given *timeout*, determined by averaging over a set of preliminary tests.

Analysis factors that we adopted are:

- F_1: participant expertise, derived from the questionnaire (novice, intermediate or expert)
- F_2: test category (with or without assistance)
- F_3: test number

We use the Analysis of Variance (ANOVA) to determine the influence of these factors on the average of these variables. Analyses were performed using the free software TANAGRA[3].

5.3.1 Non-immersive evaluation results

Assistance influence A first indication is given by the low number of failure for tests with assistance compared to tests without assistance: 3 versus 10 (Fig. 10(a)). This difference is significant ($F = 6, 168675, p < 0.01$).
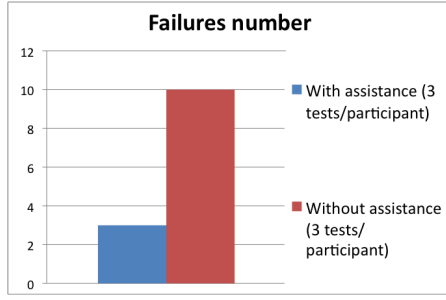
Then, the average number of users' manipulations exceeding the constraints ($N_b\_D = N_b\_dX + N_b\_dY + N_b\_dV$) is 43.06 for test without assistance, while it is 15.42 for test with (Fig. 10(b)). This difference is highly significant ($F = 12.61, p = 0.0007 << 0.01$). By studying in detail this variation, it seems more evident for volumic constraints exceeding movements ($F = 8.52, p = 0.004 << 0.01$) than for angular constraints exceeding movements on the X axis ($F = 4.21, p = 0.04 > 0.01$) or on the Y axis ($F = 2.36, p = 0.12 > 0.01$).

The average time T is decreased with assistance : $37.06s$ versus $53.42s$ (Fig. 10(c)). The standard deviation is lower: $26.55s$ with assistance against $31.43s$ without assistance. This difference is marginally significant ($F = 5.21, p = 0.02$), although the execution speed of the task is not a priority.
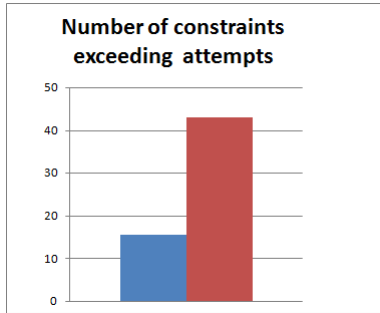
These first results show that:

1. The presence of VF allows the participants to complete the tasks at 91% of trials.
2. Designed VF seem to ensure their function, namely to inform about the presence of a constraint and enforce it. The participants have a quick perception of the model state (exceeding or not a constraint) and so identify very quickly what manipulation led to a probable exceed. Thus, participants have less hesitation to perform the task and do not persist
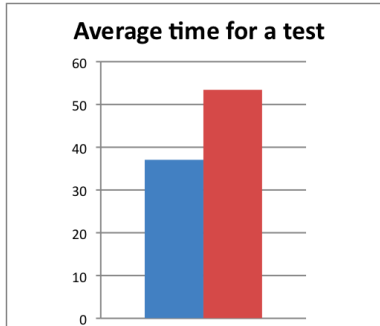
---

[3] http://eric.univ-lyon2.fr/ ricco/tanagra/fr/tanagra.html

Failures number

(a) Total number of failures.

Number of constraints exceeding attempts

(b) Average number of constraints exceeding movements per test.

Average time for a test

(c) Average completion time per test.

Fig. 10 Non-immersive evaluation results ($n = 11$, *total tests* $= 66$).

in repeating the same manipulation. That reduces N_b_D.

3. Volume guide is the most used, and this could be explained by the difficulty to perceive the three dimensional model through a monoscopic display. Users are able to manage the angular constraint because it is local to the manipulated cylinder, whereas the volume constraint is related to the whole model and is potentially exceeded by a distant cylinder.

Expertise influence Furthermore we observe that the expertise level (novice, intermediate, expert) of participants influence the average time T ($F = 9.37, p = 0.0002 << 0.01$) with a standard deviation almost equivalent to the three levels. Consequently, participant expertise influences the number of failures. But we note that the expertise determines the number of validated areas in a test N_b_Z ($F = 4.53, p < 0.01$). This means that even if in failure, an expert can validate, on average, 2.8 areas while a novice validates 2.2.

However, the expertise influence on the total number of constraints exceeding movements N_b_D is not highly significant ($F = 1.96, p = 0.14$), even if the average is 13.5 for an expert, 16.4 for an intermediary and 26.9 for a novice. This confirms that obeying to constraints does not depend on any particular knowledge in interaction with 3D objects but on expertise in the model design. We find the same results for all participants. Figure 11 illustrates the decrease in the average number of constraints exceeding movements with assistance, while N_b_D remains relatively high and irregular without assistance.
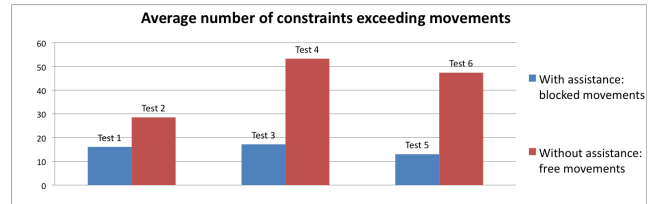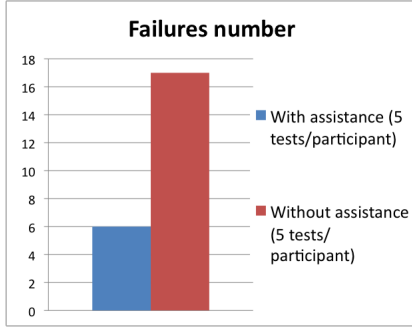

Fig. 11 Non-immersive evaluation results ($n = 11$, *total tests* $= 66$). Evolution of the average number of constraints exceeding movements for all participants per test.

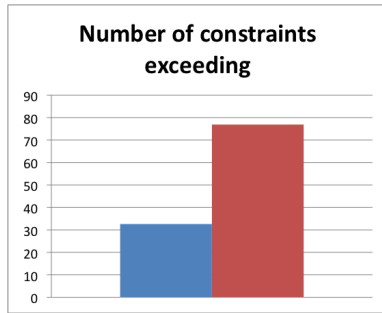### 5.3.2 Semi-immersive evaluation results

Assistance influence We keep the same analysis order as for the first evaluation. First, the failure number is greatly influenced by the assistance: 6 with vs. 17 without ($F = 6.76, p = 0.01 <= 0.01$) (Fig. 12(a)). This observation is coherent with the previous one; it confirms that even in semi-immersive environment, the assistance reduces the failures number.

Assistance also highly influences the average of constraint exceeding movements N_b_D: 32.61 with assistance vs. 76.9 without($F = 24.34, p = 0.000003 << 0.01$) (Fig. 12(b)). Unlike the first evaluation, this difference is highly significant for volume constraint ($F = 12.71, p = 0.0005 << 0.01$) but also for angular constraints (on the X axis ($F = 12.94, p = 0.0004 << 0.01$) and Y axis ($F = 11.65, p = 0.0008 << 0.01$)). We assume that the semi-immersive devices improve visual

perception of the participants, allowing them to recognize both the local and global constraints. This confirms that the global constraint (i.e. volume) was improperly perceived by participants with non-immersive system. The test category influences the average time T: $34.96s$ vs. $50.38s$ ($F = 10.4, p = 0.001 < 0.01$) (Fig. 12(b)).
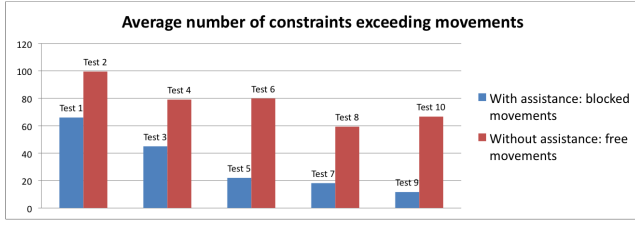
Expertise influence Concerning the relation between participant results and expertise level (novice, intermediate and expert), the conclusions are:

- The expertise level of individuals influences T ($F = 7.34, p = 0.0009 << 0.01$), with a standard deviation almost equivalent to the three expertise levels. Indeed, mastering 3DI techniques and devices simplifies the accomplishment of the task and consequently, reduces the execution time.
- The influence of expertise on the average number of failures is not significant ($F = 2.68, p = 0.07$), as well as the number of validated areas in each test N_b_z ($F = 1.92, p = 0.15$). This probably results from the naturalness of 6-DOF manipulation device, which allows novices to succeed in a test (even if it takes much more time), contrary to keyboard interaction.
- As well as non-immersive evaluation, the expertise influence on N_b_D is not very significant ($F = 2.92, p = 0.05$). N_b_D for experts is 38.22, 63 for intermediates and 63.05 for novices, with a standard deviation equal for all (roughly equal to 52). Semi-immersive experiments confirm that the constraints do not depend on any particular knowledge in 3DI.

The use of our assistance model improves the execution time and decreases the number of failures regardless of the expertise level of participant. Figure 13(a) illustrates on one hand, a decrease in N_b_D with assistance, while it remains relatively high without assistance. On the other hand, we observe a decrease of the total time T (Fig. 13(b)). For these two parameters, N_b_D and T, a small learning curve emerges during tests with assistance. Thus, we can see that learning the approach of constraints-based 3DI and the assistance model improves the results of a participant, from one test to another. However, it is important to note that the regular alternation between tests with and without assistance may have influenced learning. Moreover, since the initial conditions of each experiment are the same (except point of view), participants might have learned progressively manipulation to do to validate the experience. This could also explain the slight irregular decrease of N_b_D and T for test without assistance.

5.3.3 Comparison between non-immersive and semi-immersive assistance

To compare results of non-immersive with desktop interactions (noted NI) and semi-immersive with VR interactions (SI) evaluations, we followed the same evaluation scenario and the same initial conditions for each test. Ten participants participated to both experiments.

(a) Total number of failures.

(b) Average number of constraints exceeding movements per test.

(c) Average completion time per test.

Fig. 12 Semi-immersive evaluation results ($n = 12$, *total tests* $= 120$).

(a) Evolution of the average time.

(b) Evolution of the average exceeding number per test.

Fig. 13 Semi-immersive evaluation results ($n = 12$, *total tests* $= 120$).

Tests with assistance N_b_D is identical for both environments. So we can note that the CTT model is as effective for non-immersive environments as for semi-immersive environments. This confirms the genericity of the model and its independence from the used devices. T is $39.72s$ in semi-immersive environment (standard deviation 22.91) and slightly lower in non-immersive one, $37.69s$ (standard deviation 27.43). Our analysis is that the selection is a little more difficult and time-consuming with the VR interaction (Flystick device and virtual ray technique) than with the mouse (2D picking) because of the increased DOF and the lack of physical support for the hand. It also adds an extra time for the navigation task.

N_b_D is 16.96 for NI and 37.33 for SI. The number of constraints exceeding movements has almost doubled. Here again we think this is due to the increased DOF with the VR device and technique for the manipulation task. Indeed, the user acts on three rotation axes while the keyboard allows a single rotation axis (up-down arrows correspond to an axis and left-right to another).

We can make two statements:

1. VF are used and assist the participants in the two evaluations.
2. The difference is great between assistance to the NI and SI manipulation. NI interaction is already limited by the device and the manipulation technique, while the latter offers more freedom to the user, which involves a greater risk of error.

Table 4 Gains of adding assistance VF for each parameter according to the experimental platform.

Parameters	Non-immersive	Semi-immersive
Failures number	72.22%	61.54%
Average number of constraints exceeding movements	63.89%	55.53%
Average time for a test	32.48%	23.72%

Responses to the questionnaire of the second evaluation are coherent with what we previously suspected. To the question "do you think the assisted SI interaction mode is better than in NI?", 70% of the participants answered "yes" and 30% responded "probably yes". This seems natural because the participants judged VR interaction more intuitive and less constricting, so they try new changes, which leads to greater number of constraints exceeding attempts. Therefore, users are more reassured by the given assistance. This result justifies the need of our assistance model for the constraints-based interaction, especially for VR interactions.

Comparison Now we want to clearly quantify the performance gain of moving from non-immersive to semi-immersive 3DI of our CTT model. We focus on assisted and non-assisted tests of each experiment and we calculate the performance gain for each analysed parameter (number of failures, number of constraints exceeding movements and execution time of a test). Table 4 presents gains on all parameters and for each evaluation. With these global parameters, it is difficult to determine a possible comparison between the contribution of the assistance model in NI and SI because several additional factors come into play, as the complexity of the 3D selection, adding the navigation task or the system control *via* the flystick for the second evaluation. However, we can observe that:

1. For both experiments, the assistance model reduces more than 50% the average number of failures and the average number of constraints exceeding movements.
2. The gain on the average time for performing a test is also comparable to the experience with NI and SI.
3. Although the assistance model seems to provide a performance gain greater for NI, it is nonetheless for SI. Despite the additional navigation and selection time, the gain is almost equal for both platforms.

6 Conclusion

We have developed a new framework Constraint-Tool-Task (CTT) to assist 3DI in constraints-based VE, thus

to guarantee the integrity of users' experiences. The assistance tools we have formalized are Virtual Fixtures. They are generic and multimodal. Their structure allows a reference to a specific global or local constraint, a 3DI task, and a rendering on visual, audio and haptic modalities. We have implemented our CTT framework in the case of the *in virtuo* Human-Molecule interaction. The aim of this application is to propose a new way to analyse a chromosome by manipulating its 3D model. It requires the integration of three specific physico-chemical constraints in the interaction system. Each constraint was associated to a VF in accordance with our formalism. Then we compared our assistance model to conventional (non-assisted) 3DI, both in non-immersive and semi-immersive environments. The two evaluations of our approach (objective, based on the parameters of time, number of failures and the number of constraints exceeding movements; subjective, based on questionnaires) are coherent and show the gain brought by the assistance we have designed. VF gives better information on constraints, and make the participant more effective. Moreover, their unified structure allows clear, logical and adaptable implementation.

Future works are the formalisation of haptic feedback. Indeed, it may be interesting to naturally "feel" the limits imposed by the constraints, e.g. a deformation, a collision or an unreachable zone. First studies show that the CTT model can handle haptic VF by adding to the modality field a force profile and a direction function linked to the primitive. This will lead us to evaluate again the system on a VR platform to identify more precisely the contribution of multimodal and immersive assistance.

References

1. Abbott J, Marayong P, Okamura A (2007) Haptic virtual fixtures for robot-assisted manipulation. Robotics Research pp 49–64
2. Bowman D (1999) Interaction techniques for common tasks in immersive virtual environments:–design, evaluation, and application/–by douglas a. bowman. PhD Thesis, Georgia Institute of Technology
3. Bowman D, Hodges L (1995) User interface constraints for immersive virtual environment applications. Graphics, Visualization, and Usability Center Technical Report GIT-GVU-95-26
4. Bowman D, Hodges L (1997) An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. Proceedings of the 1997 symposium on Interactive 3D Interactive 3D graphics pp 35–ff
5. Bowman D, Koller D, Hodges L (1998) A methodology for the evaluation of travel techniques for immersive virtual environments. Virtual Reality
6. Bowman DA, McMahan RP (2007) Virtual reality: How much immersion is enough? Computer 40(7):36–43
7. Brooks T, Ince I, Robotics H (1992) Operator vision aids for telerobotic assembly and servicing inspace. IEEE International Conference on Robotics and Automation, Proceedings 1:886–891
8. Calderon C, Cavazza M, Diaz D (2003) A New Approach to the Interactive Resolution of Configuration Problems in Virtual Environments. Smart Graphics: Third, Springer Berlin Heidelberg pp 112–122
9. Castet J, Florens J (2008) A virtual reality simulator based on haptic hard constraints. Haptics: Perception, Devices and Scenarios, Lecture Notes in Computer Science 5024:918–923
10. Cooper S, Khatib F, Treuille A, Barbero J, Lee J, Beenen M, Leaver-Fay A, Baker D, Popovic Z, Players F (2010) Predicting protein structures with a multiplayer online game. Nature 466(7307):756–760
11. Essabbah M, Otmane S, Hérisson J, Mallem M (2009) A new approach to design an interactive system for molecular analysis. Springer, Human-Computer Interaction Interacting in Various Application Domains 5613:713–722
12. Essabbah M, Otmane S, Mallem M, Herisson J (2009) Spatial organization of dna: From the physical data to the 3d model. IEEE/ACS International Conference on Computer Systems and Applications pp 880–883
13. Férey N, Delalande O, Grasseau G, Baaden M (2008) A vr framework for interacting with molecular simulations. ACM symposium on Virtual reality software and technology pp 91–94
14. Férey N, Nelson J, Martin C, Picinali L, Bouyer G, Tek A, Bourdot P, Burkhardt J, Katz B, Ammi M, Etchebest C, Autin L (2009) Multisensory vr interaction for protein-docking in the corsaire project. Virtual Reality 13(4):257–271
15. Fernando T, Murray N, Tan K, Wimalaratne P (1999) Software architecture for a constraint-based virtual environment. In: Proceedings of the ACM symposium on Virtual reality software and technology, ACM, New York, NY, USA, VRST '99, pp 147–154
16. Gibson JJ (1977) The Theory of Affordances. In Perceiving, Acting, and Knowing. Robert Shaw and John Bransford

17. Gillet A, Sanner M, Stoffler D, Olson A (2005) Tangible interfaces for structural molecular biology. Structure 13(3):483–491

18. Guébert C, Duriez C, Grisoni L (2008) Unified processing of constraints for interactive simulation. Proceedings of VRIPHYS

19. Heyd J, Birmanns S (2009) Immersive structural biology: a new approach to hybrid modeling of macromolecular assemblies. Virtual Reality 13(4):245–255

20. Jacoby R, Ferneau M, Humphries J (1994) Gestural interaction in a virtual environment. Proceedings of SPIE 2177(355)

21. Kalawsky R (1996) Exploiting virtual reality techniques in education and training: Technological issues. Tech. rep., Advanced VR Research Centre, Loughborough University of Technology, URL http://www.agocg.ac.uk/reports/virtual/vrtech/title.htm, advisory Group on Computer Graphics (AGOCG)

22. Kuang AB, Payandeh S, Zheng B, Henigman F, MacKenzie CL (2004) Assembling virtual fixtures for guidance in training environments. Haptic Interfaces for Virtual Environment and Teleoperator Systems, International Symposium on 0:367–374

23. Marayong P, Li M, Okamura AM, Hager GD (2003) Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. In: Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14-19, 2003, Taipei, Taiwan, IEEE, pp 1954–1959

24. Marcelino L, Murray N, Fernando T (2003) A Constraint Manager to Support Virtual Maintainability. Computers & Graphics 27(1):19–26

25. Mine M (1995) Virtual environment interaction techniques. UNC Chapel Hill Computer Science Technical Report TR95-018

26. Mine M, Jr FB, Sequin C (1997) Moving objects in space: exploiting proprioception in virtual-environment interaction. Proceedings of the 24th annual conference on . . .

27. Oakley I, Adams A, Brewster S, Gray P (2002) Guidelines for the design of haptic widgets. In Proceedings of British HCI, British Computer Society pp 195–212

28. Otmane S, Mallem M, Kheddar A, Chavand F (2000) Active virtual guides as an apparatus for augmented reality based telemanipulation system on the internet. IEEE Annual Simulation Symposium, 2000 (ANSS 2000) Proceedings pp 185–191

29. Ouramdane N, Otmane S, Davesne F, Mallem M (2006) Follow-me: a new 3d interaction technique based on virtual guides and granularity of inter-

action. Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications pp 137–144

30. Picon F, Ammi M, Bourdot P (2008) Haptically-aided extrusion for object edition in cad. Haptics: Perception, Devices and Scenarios pp 736–741

31. Pierce J, Forsberg A, Conway M, Hong S (1997) Image plane interaction techniques in 3d immersive environments. Proceedings of the 1997 symposium on Interactive 3D graphics pp 33–ff

32. Poupyrev I, Billinghurst M, Weghorst S (1996) The go-go interaction technique: non-linear mapping for direct manipulation in vr. Proceedings of the 9th annual ACM symposium on User interface software and technology pp 79–80

33. Prada R, Payandeh S (2009) On study of design and implementation of virtual fixtures. Virtual Reality 13(2):117–129

34. Ren J, Zhang H, Patel R, Peters T (2007) Haptics constrained motion for surgical intervention. Health Technology and Informatics pp 379–384

35. Rosenberg L (1992) The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments. Technical report, No A054292, USAF Amstrong

36. Simard J, Ammi M (2011) Haptic interpersonal communication: improvement of actions coordination in collaborative virtual environments. Virtual Reality pp 1–14

37. Smelik R, Galka K, De Kraker KJ, Kuijper F, Bidarra R (2011) Semantic constraints for procedural generation of virtual worlds. Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, ACM p 9

38. Smith G, Stuerzlinger W, Salzman T (2001) 3d scene manipulation with 2d devices and constraints. Graphics Interface pp 135–142

39. Sternberger L, Bechmann D (2005) Deformable ray-casting interaction technique. IEEE Young Virtual Reality Conference

40. Sweller J, Ayres P, Kalyuga S (2011) Cognitive Load Theory. Springer