IPT-EGVE Symposium (2007)
B. Fröhlich, R. Blach, and R. van Liere (Editors)

Supervision of Task-Oriented Multimodal Rendering for VR Applications

G. Bouyer, P. Bourdot and M. Ammi

VENISE transversal action, LIMSI-CNRS, Université Paris-Sud, France

Abstract

This article addresses the question of integrating multimodal rendering in Virtual Reality applications. It exposes first the interest of multimedia intelligent systems to improve human activity in Virtual Environments. Then it details the conception of a software module in charge of supervising multimodal information rendering, depending on the interaction and its context. From existing psychophysical studies and concrete applications, we propose a model, an architecture and a decision process. Finally a first implementation is presented to validate the core of the simulator and show the adaptability of its knowledge base.

Categories and Subject Descriptors (according to ACM CCS): D.2.2 [Software Engineering]: Design Tools and Techniques [User Interfaces] ; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems [Artificial, augmented, and virtual realities]

1. Introduction

Virtual Reality (VR) is a discipline which allows human users to perceive and manipulate numerical environments in an immersive, pseudo-natural and real-time way. Right from the start and the Sensorama Simulator of Morton Heilig in 1962, VR applications tried to combine multiple sensory stimulations. In the course of technological advances, VR systems have integrated new interaction interfaces (speech, gesture, visual, auditory, haptic, etc.) to make the most of all human sensori-motor capacities. These systems are called "multimedia", "multisensorial" or "multimodal". The last type is also referred to as "multimedia intelligent system".

Such "multisensorial" systems have been developed on one hand to improve the realism of Virtual Environments (VE), and on the other hand to enhance speed, efficiency and comfort in numerous tasks. We call the last objective the "task-oriented" approach. This approach is based on the Modal Specific Theory [Fri74]. This theory states that each sensorial channel has a unique method of information processing and is suited for a certain sort of stimulus. For example, vision is a spatial channel, capable of interpreting spatial relationships. Audition is not really efficient for 3D tasks, but is useful to perceive spatial information that are not located in the field of vision. Moreover it is a very efficient channel for analysing temporal phenomena. Haptic sense is special as it requires active perception, i.e. movement, to give both temporal and spatial cues. Sensorial modalities are the various means of transmitting information through the channels. So according to the MST theory, each modality has its own advantages and drawbacks and is appropriate to a certain type of information. The main purpose of multisensorial systems is to find the more efficient and appropriate mapping between a piece of information and a modality (or a combination of several modalities).

However, the mapping rules between modalities and information are not valid in all situations ; the rendering should depend not only on the data to be communicated but also on the context of the interaction (user, environment and system) [NC93]. To operate properly, "multimodal" systems should integrate an intelligent artificial manager to adapt multisensorial interaction to various users, to be flexible to different contexts of activity. This manager is based on a knowledge representation of the human channels, of the operator objectives, of the rendering capacities of the technical architecture, etc. One of the main issues of VR multimodality is to model and develop such a system for various applications.

2. Previous work

In this section we review some task-oriented researches on either multisensorial or multimodal rendering designed to improve the activity of the users.

In the learning and training domain, ScienceSpace [SDL96] has been designed to improve pupils' understanding of physical laws: movement (Newton), electrostatic forces (Maxwell) and molecular structures (Pauling). In these three applications, different visual modalities are relevantly used to perceive some phenomena invisible in the real world. In the same domain, Infed [IBL*99] combines visual and haptic renderings to display multidimensional and massive data for the analysis of electromagnetic fields and fluid mechanics. For scientific exploration and analysis applications, McLaughlin [MO97] uses force feedback to assist seismic interpretation of geophysical data. This approach helps users feeling elements that are masked by the visual representation. In [BZ00] Barass uses the auditory signals on the Geiger meter metaphor to render some information related to oil exploration. In this application the user observes the visual model of the oil well and listens to the characteristics of some data at the same time. Fitch [FK94] uses the same approach to render physiological parameters like breath rate, body temperature or heart rate for medical purposes.

All these examples present successful task-oriented VR applications which try to make the most of qualities and availability of feedbacks to create possibly non-realistic but efficient interactions. Complementarity or redundancy between several stimuli are used to increase the meaning in human-computer communication. However, these relevant multisensorial mappings are designed for precise situations and are not associated with adaptative systems that observe the context, adapt to the environment and are able to choose other suitable multisensorial renderings.

Whereas there are numerous works on output multisensoriality in HCI or VR, for instance on dynamic generation of graphics or speech depending on the information, there are few studies on adaptive multimodal rendering on the context (user, system, environment). A recent work on this issue is the Eloquence platform [RBVB06] for the specification, the simulation and the execution of output multimodal systems. Two applications are tested within this framework: the simulation of an incoming call in an intelligent mobile phone and a task of designating a target on the ground in a fighter plane cockpit. Unfortunately, VR systems raise specific problems that are not handled in other HCI domains. In particular, VR applications must handle spatial and temporal representations of the user in order to generate adapted 3D presentations (for instance a stereoscopic rendering) and to allow mutual references between different devices. These differences lead to different models and contexts of interaction.

The objective of the present article is to expose our view of a multimodal system. It details the conception of an "artificial multimodal supervisor" to manage multisensorial interactions in VR applications. We first explain our conception method, directed by two loops of evaluation: a psychophysical one and an ergonomic and application one. Then we develop a model of interaction suitable for multimodal knowledge representation. This model is implemented in our supervisor, which handles the knowledge rules, considers the context and cooperates with the application and the user. Finally we discuss our first results and our perspectives.

3. Method

Humans are at the heart of VR applications. They are immersed in VE. That is why we think psychophysical experiments are key to relevant multimodal rendering. First, results of existing studies on perception and modalities combination are integrated in the mapping rules used by our supervisor. Secondly we intend to identify cross-effects between modalities, inadequate combinations or predominant influencing conditions. Moreover, VR applications are intended to be used by "real users", so designers must respect users' needs and desires, and the conception must follow ergonomic and usability methods.

That is why we have chosen to obey to two loops of evaluation during our conception of a multimodal supervisor (figure 1). The organisation of our article follows the same path: after presenting existing psychophysical studies and the concrete applications and interactions aimed at by our work, we detail how these analysis have been integrated in our supervisor.

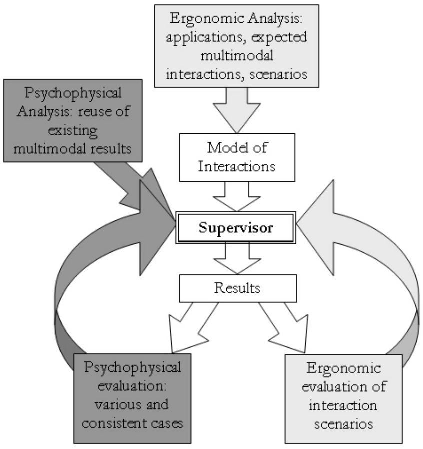

Ergonomic Analysis: applications, expected multimodal interactions, scenarios
Psychophysical Analysis: reuse of existing multimodal results
Model of Interactions
Supervisor
Results
Psychophysical evaluation: various and consistent cases
Ergonomic evaluation of interaction scenarios

Figure 1: *Two loops of evaluation lead the conception of our multimodal supervisor.*

3.1. Psychophysical inspiration

The main class of experiments we considered are illusions. Indeed, illusions occur when the perception or the mind makes us believe in things that are not real. This is precisely the essence of VR. Haptic illusions are especially interesting. For instance, Robles-De-La-Torre and Gosline have shown that we can create illusions of shapes (bumps, hollows [RDLTH01] and curvature [GTB01]) on a plane by mixing force cues with geometric cues. Lécuyer [LCK*] has even created a pseudo-haptic illusion using only visual rendering. Experiments are also conducted on the subject of audio-haptic crossmodal effects, for example during perception tasks of texture, roughness or taps.

These experiments are conducted in various environments and for various tasks and objectives. We try to draw our inspiration from their results in order to create VR interactions which respect human characteristics and behaviors. We want to reuse illusions and cross-modal effects either to create new renderings or to avoid combining incompatible modalities.

3.2. Applications analysis

Two domains were selected as representative for VR multimodal applications: Fluid Mechanics and Molecular Docking of biological complexes. Firstly, both aim at improving the manipulation or the exploration of data. This raises problems of mapping between information and modalities. Secondly, there can be unpredictable situations that make imprecise, inadequate, difficult or even impossible the rendering planned by the conceptor.

For Fluid Mechanics data exploration (figure 2), information to be processed consist of vectors representing physical quantities (velocity, pressure, temperature, scalar gradients, etc.) sampled on a three-dimensional mesh. The major difficulty is to simultaneously observe about ten parameters or quantities of four dimensional (space and time) data. If the visual immersion appears to be the obvious method to gain a perception of the numerical simulation of a flow, auditive perception can advantageously be used to perceive phenomena of intermittent nature or strongly localised (intermittency of turbulence, ruptures of structures resulting from the swirling stretching). Moreover, visual immersion admits limitations primarily related to the overload of scenes which can hide required information (too many iso-surfaces which limit the depth of visualization, for example). Masked information should then require a rendering on another sensorial channel, or if it is not appropriate the system should remove previous visualizations. Another delicate situation would be raised if the user's attention is disturbed or turned toward another target while manipulating a data. For example, while haptically moving a cutting plane around an interesting position, the operator can turn towards colleagues: it would be interesting to test whether the system should block the device to save his/her current position or not.

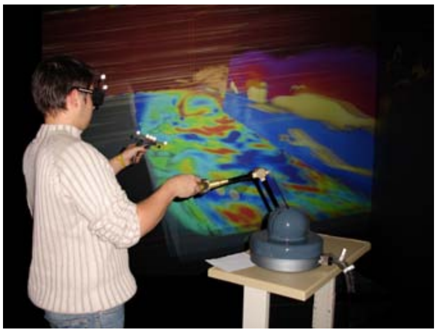


Figure 2: *Haptic manipulation of a cutting plane for Fluid Mechanics iso-surface exploration.*

One of the main issues studied in Molecular Bioinformatics is the protein docking problem. Current works separately use visualization softwares to observe the proteins topologies and characteristics (figure 3) and powerful calculators to score the best possible configurations between two proteins. A promising approach to this problem is to merge the biologist into the loop of simulation, to give him the possibility to naturally interact with proteins (by using for example haptic devices). He/she could select some valid configurations before the calculation, and therefore avoid heavy and useless tests of irrelevant positions. On the other hand, the system could present useful information (on proteins or on the variables resulting from prediction calculations) in visual, haptic, or auditive form to cooperate with the expert. Once again one of the problems is the simultaneous presentation of multiple pieces of information. For example, haptic rendering seems to be well suited to render the electrostatic interaction between the proteins, but should not be used if information about hydrophobia or topological complementarity already monopolize this modality.

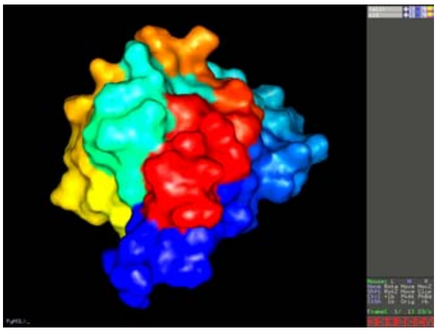


Figure 3: *Molecule surface 3D visualization using PyMOL.*

Our view is that such applications need a rendering supervisor. A first step towards such a manager is to build a model of possible interactions, to precisely describe tasks, user expectations and multimodal capacities.

4. Interaction Model

4.1. Tasks

From the analysis of various VR applications and especially the two previous ones, we have set up a model of interactions. Each interaction is represented by three elements: the task that is performed, the parameters specifying the task and the data concerned by the task (figure 4).

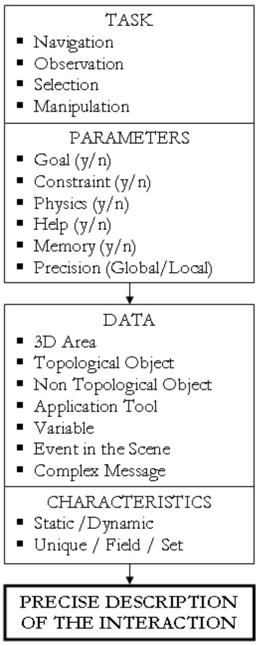


Figure 4: *Our model of interactions: task, parameters and data.*

In the scope of this article we will detail only some of these elements. Concerning the four types of task, "observation" is the perception and the processing of data coming from the VE, without modifying them, whereas "manipulation" is the displacement or the modification of these data. Three parameters are especially interesting for our applications: "constraint" indicates that the system must prevent the user from interacting in some 3D area of the VE; "physics" specifies that the VE must obey to physical laws, especially the non-interpenetration of the objects; "help" points out that the system must collaborate with the user, for example to find a target or follow an optimal path.

This model allows us to represent numerous concrete interactions. For example, in Fluid Mechanics, when one explores some iso-surface with a proxy, one can decide that the system must manage physical collisions between the proxy and the iso-surfaces. This interaction is modelled by:

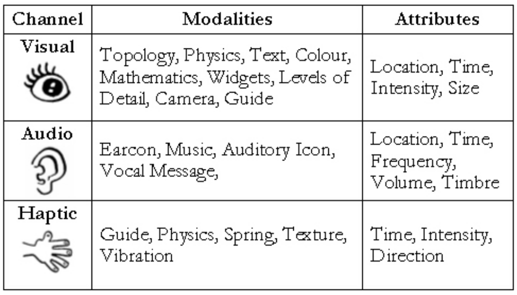
```
task = manipulation
parameter = physics
data = event in the scene = contact
```

The strong point of this model is that different interactions can be represented with exactly the same elements, leading to a same supervision, thus producing the same rendering for given conditions. For example, in protein docking, if the user wants to manipulate a protein and feel the physical topological complementarity with another protein, the description is identical to the previous one, even if the manipulated objects are different. On the contrary, if the biologist wants to feel an electrostatic interaction and be guided towards the best configuration, the model is:

```
task = manipulation
parameter = help
data = float variable
```

4.2. Multimodal Rendering

In order to identify multimodal rendering possibilities, again an analysis of applications and existing VR systems was done. Our taxonomy distinguishes between: human sensorimotor *channels* (e.g. vision, audition, kinesthetic and tactile systems); technological *media* which are the devices corresponding to the channels (screens, projectors, speakers, haptic devices, etc.) and *modalities* that are the various means to transmit information through the media. For example, on a screen, we can see either text, or shapes, or colours, etc.



Channel	Modalities	Attributes
Visual	Topology, Physics, Text, Colour, Mathematics, Widgets, Levels of Detail, Camera, Guide	Location, Time, Intensity, Size
Audio	Earcon, Music, Auditory Icon, Vocal Message,	Location, Time, Frequency, Volume, Timbre
Haptic	Guide, Physics, Spring, Texture, Vibration	Time, Intensity, Direction

Figure 5: *Typical VR modalities and their attributes.*

We have identified in figure 5 a list of practical modalities that are available in nowadays VR installations and that we use in our examples. Each modality can be further specified by several attributes. There is for instance the temporal attribute which is common to all modalities: starting time of rendering, duration time, etc. We can see that there are a lot of possibilities for one modality, but the combinatorial choices become even larger when it comes to combine several renderings. Vernier [VN00] suggests a complete categorisation for the combination of two messages, based on five temporal relationships suggested by Allen [All83]: precedes, meets, overlaps, during and equals. In the current study we restrict ourselves to the "equal" relationship: if there are several modalities, they are rendered at and during the same time.

Of course, these task parameters, data and modalities are not exhaustive. However, they allow to precisely describe multimodal interactions and represent a trustworthy and extensive panel of what users can do in VR applications. Moreover they allow to build a multimodal rendering supervisor based on a model with well determined components.

5. Multimodal Supervisor

In front of so many interactions, so many possible renderings and so many unpredictible situations, the authors' view is that multimodal VR applications need an intelligent agent in charge of supervising the rendering. This "multimodal supervisor" would be in charge of: handling a knowledge base of logical mapping rules between different types of tasks, parameters and data and available multimodal representations; analysing the context thanks to observer modules and balance the mapping rules depending on the situation; controlling, replacing, modifying or tuning the rendering if it does not fit to the interaction and/or the context; communicate with the application and cooperate with the user. Our objective is to keep a global control of the interactions, while ensuring that the communications between the user and the system are consistent and respond to the same logical, ergonomic and psychophysical recommendations.

5.1. Specifications

On the basis of our objectives, the analysis of previous HCI and psychophysical studies, the targeted applications and the users needs, we have set up a list of crucial specifications for our multimodal supervisor.

1. The supervisor should be generic, i.e. independent of the application and the devices, adjustable to new environments, new tasks, new users and adaptable to new multimodal knowledge.
2. The system obeys the user, so the supervisor must be subject to the user's commands and choices, when it is possible and safe.
3. Besides knowing what is to be represented (tasks, parameters and data) and the representation (modalities), we need to know the set of variables influencing the mapping (context) and state the rules of mapping (or distribution). This is done by analysing the applications and carrying out psychophysical experiments.
4. The supervisor should be a tool to evaluate the use of multimodality in VR applications. It should record the interactions in order to measure the choices of the users and the level of acceptability and usage of multimodality.
5. Last but not least, the multimodal supervisor should be practically used, that is accepted by developers and users.

5.2. Architecture

The architecture of the multimodal supervision is detailed in figure 6.

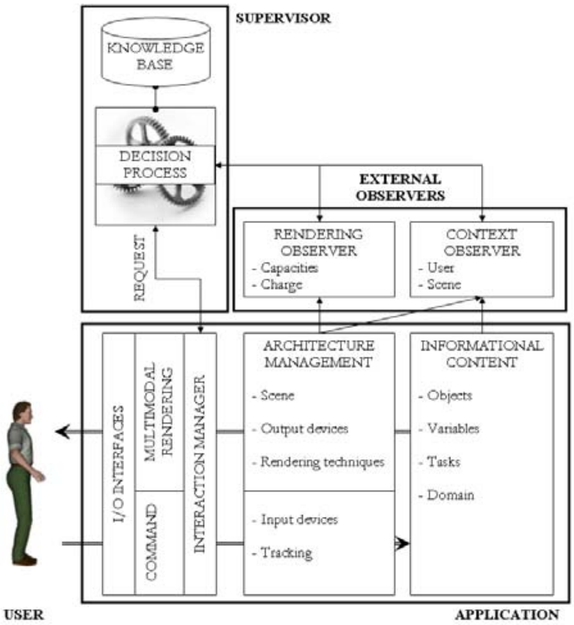


Figure 6: *Architecture of multimodal rendering supervision in a VR application.*

The user interacts with the informational content of the application through input and output interfaces. Interactions are divided between commands from the user to the system and rendering from the system to the user. The informational content contains the global scene, the objects, the meaningful variables, the representation of the user. It is handled by a hardware architecture which manages the scene and the devices.

Interactions are designed by the developer of the application, who chooses the modalities and their attributes for all the possible tasks. However, he/she can not foresee every possible situation. That is why the application must propose the potential interaction to the supervisor which is then in charge of validating, modifying or specifying it. In accordance with our model, each requested interaction is represented by a task, some task parameters and a piece of data.

The decision process of the supervisor is detailled in section 5.3. It first depends on the knowledge base of mapping rules between the task, the parameters, the data and the available modalities. Then it depends on the context. Contextual elements are tracked in real time by two external modules: one observes the state of the scene by translating various state variables (tracking data, etc.) into meaningful parameters (in particular where the user is, what he is doing and how the scene is organised); the other one records the current rendering capacities of the application (number of simultaneous interactions, available modalities and charges of the media). When the user has decided to operate with specific modalities, the supervisor has to take into account this superceding command: this means it must validate the interaction, even if it is not the best choice, unless it is impossible or dangerous. The result of the request is one or two modalities with their parameters. The requested interaction is updated, concretely rendered through the devices and recorded for future evaluation.

5.3. Decision process

The process between the request to the supervisor and the result of the decision is detailed in figure 7. The process, the knowledge base and the elements of the context are all programmed in Prolog, in order to benefit from logical languages proof capacities.

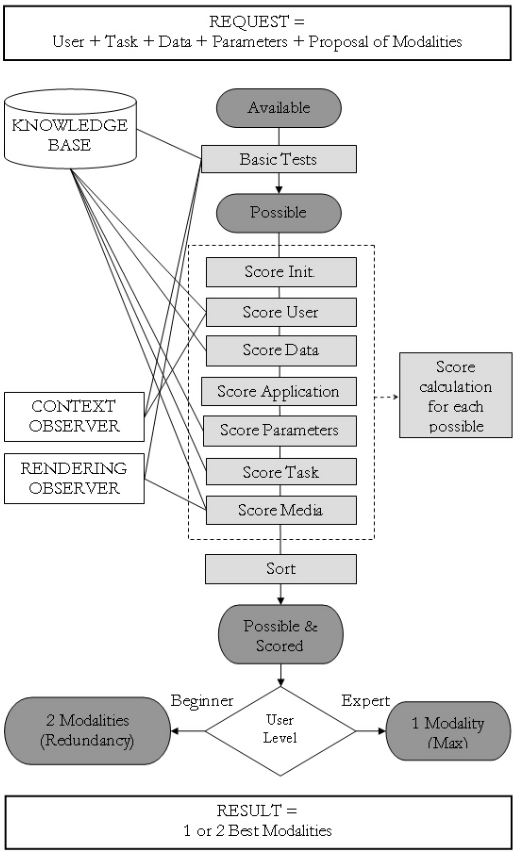


Figure 7: *Sequence of filters and scores leading to the choice of the rendering modalities.*

First, each component of our model of interactions must be implemented.

The user is identified by a name and characterized by a list of maximum three sensori-motor channels (visual, audio and haptic) and a level of expertise:

```
user(<UserID>, <Channel List>, <UserLevel>).
```

This allows to adapt to possible handicaps by using or not modalities depending on the required sensorial channel.

The available media are represented by a type (visual, audio and haptic) and a charge, i.e. the number of modalities currently being transmitted through the media:

```
media(<MediaType>, <Charge>).
```

The charge is updated at the beginning and at the end of each concrete rendering. This update is done by the rendering observer.

The available modalities (implemented in the application) are described by a name and an associated medium:

```
mod(<Modality>, <MediaType>).
```

The requested interaction is described by an ID, a task, the parameters of the task and the data concerned by the task. Each of these elements is represented by a type, which has been identified in the model of interactions (section 4).

```
task(<InteractionID>, <TaskType>).
param(<InteractionID>, <ParamType>).
data(<InteractionID>, <DataType>).
```

The first example cited in section 4.1 becomes:

```
task(0, manipulation).
param(0, physics).
data(0, contact).
```

Apart from the dynamic predicates which represent the interactions, the Prolog base is composed of static knowledge and rules. The principle of the decision process is the manipulation of lists of modalities: starting from the list of all available modalities in the application, the supervisor tries to come to a selection of one or two suitable modalities. It first filters available modalities with basic tests, depending on user's capacities and media's charges. Then it elaborates a score for each possible modality. This calculation relies on a knowledge base of lists of favourite modalities for each element which could influence the rendering: the proposal of the application, the type of data, the types of parameters, the type of task and the user's preferences. Starting at 0, the score is incremented or decremented each time the modality is a member of a list of favourite modalities (the order does not matter). At the beginning, the incrementation or decrementation is only of 1, but the process is designed for future adjustments: each modality in each list can receive a personnal weight to manage a priority system. Current "favourite rules" look like:

```
data_modfav(<DataType>, [(M1, W1), ...]).
```

These rules are defined by the conceptor, because at this time the system can not infer them with artificial intelligence. There is also a rule to avoid heavily charging the media.

Finally the score of each modality for the requested interaction is :

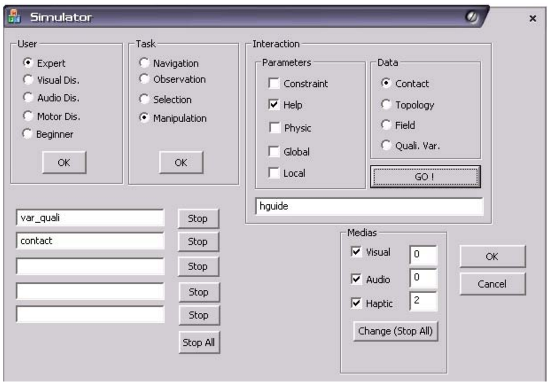
```
Score(Modality, InteractionID) =
  Score(Modality, UserID)
+ Score(Modality, DataType)
+ Score(Modality, AppliProposal)
+ Score(Modality, ParamType)
+ Score(Modality, TaskType)
+ Score(Modality, MediaType)
```

Amongst all these possible scored modalities, the final choice of rendering depends on the expertise of the user. If the operator is an expert, we consider for the time being that a single modality (maximum score) is sufficient to perform the task; if the user is a beginner (level < 3) or has disabilities, we choose to select the best two modalities, on the idea that redundancy can influence positively the efficiency and the correctness of the task.

6. Implementation

6.1. Simulator

In order to validate the above rules and to avoid application-dependent implementation, we have developped a simulator of our supervisor based on WIMP paradigms (figure 8). The simulator allows to create different profiles of users and various interactions and then test the supervisor decision. It also allows to manage some aspects of the context, such as media availability, media charges and user's focus of attention.



User: ● Expert ○ Visual Dis. ○ Audio Dis. ○ Motor Dis. ○ Beginner

Task: ○ Navigation ○ Observation ○ Selection ● Manipulation

Interaction

Parameters: □ Constraint ☑ Help □ Physic □ Global □ Local

Data: ● Contact ○ Topology ○ Field ○ Quali. Var.

hguide

var_quali

contact

Medias: ☑ Visual 0 ☑ Audio 0 ☑ Haptic 2

Figure 8: *Simulator of our multimodal supervisor.*

The visible part of the simulator is programmed with MFC (Microsoft Foundation Classes). The core of the program manipulates C++ classes and threads in order to parallelize all the requests. The Prolog parts and the Prolog-C++ communication are provided by Amzi! Prolog Logic Server. All the elements are designed and programmed to adapt to a real VR C++ application.

6.2. Results and discussion

The first step towards our supervisor evaluation is the tuning of our knowledge base. That is why our simulator was first tested on a few precise interactions. The lists of modalities in the `X_modfav` predicates are iteratively defined until the supervisor gives the expected results. Each modality has a weight of 1. This iterative process has ensured establishing rapid consistent behaviour. For example, after these preliminary settings, the interactions described in section 4.1 give the following correct results:

```
task = manipulation
parameter = physics
data = event in the scene = contact
result = [(physical haptic, 3),
          (physical visual, 3)]

task = manipulation
parameter = help
data = float variable
result = [(tactile vibration, 3)]
```

We can note that physical haptic and physical visual modalities are closely linked because if the non-interpenetration is haptically managed, visual non-interpenetration is automatically handled.

Then other concrete interactions were experimented to test if the supervisor produces consistent results given new combinations of the previous elements. For example, a useful help for physicists would be the attraction of the proxy when it is close to an iso-surface. This task is modelled by:

```
task = manipulation
parameter = help
data = event in the scene = contact
```

Each element already has its corresponding favourite rule. Four output modalities would be accepted: `haptic guide`, `visual attraction`, `audio variation` and `tactile vibration`. All these modalities give more or less natural information about the distance between the proxy and the target, so they could help the user to reach it. The result given by the supervisor is `result = [(haptic guide, 2)]`.

The first significant comment is that the decision is taken in real time. This means that the Prolog process, the communication between the C++ application and the Prolog base and the update of the context is done in real time (about 90 ms from the request to the result). This is promising for future use in VR applications which are known not to accept important latencies. The second comment is that the supervisor gives a correct answer. However, in the above example, the other three acceptable modalities have the same score of 2, along with `physical haptic` and `physical visual`. This profusion of proposed modalities can yet be reduced by some simple modifications. For instance, it is suitable to favour haptic modalities for manipulation tasks: this is quickly applied in our rules by increasing the weight of haptic modalities in the favourite list of manipulation. Similarly, we can increase the weight of the `haptic guide` when the user is expecting a help from the system. With such flexibility, our system is sure to favour `haptic guide` for each identical type of interaction:

```
result = [(haptic guide, 4)]
```

Apart from the choice of the best rendering for a given task, another essential role of our supervisor is to adapt the rendering to the context. The most common situation is the non-availability of a medium or a user's disability. For instance, we can test the previous interaction by turning off the haptic device. All haptic modalities are impossible so the result is:

```
result = [(visual attraction, 2)]
```

The visual representation of the proxy is then attracted to the visual representation of the iso-surface. Here again the supervisor gives an appropriate answer, consistent with the context.

7. Conclusions and Future Work

In this article, the problem of multimodal rendering in VR applications was adressed. We have discussed the need for multimodal rendering to enhance the quality of VR tasks. Then the design of a multimodal supervisor was presented. Starting from existing psychophysical studies and practical applications, we have modelled multimodal interactions and proposed an architecture of communication between a user, an application and a rendering supervisor. Rendering decisions are taken by scoring lists of modalities depending on the interaction and its context. Finally we have validated the core of our supervisor on concrete scenarios mixing various tasks and situations. First tests have shown that the supervisor gives correct answers in real time depending on the task, the parameters, the data and the context. Our knowledge base is also simply adjustable to manage many different tasks and needs.

Our first objective is now to fine tune the knowledge base (type of event, type of task, parameters of task, user choice) from a deep ergonomic analysis of our applications. Then statistical tools and fuzzy logic will serve to optimize the scores of the decision rules. The next step will be the implementation of the supervisor in our existing Fluid Mechanics exploration software. The major difficulty is the adaptation of the application input and output methods. The internal representation of the interactions must also be compliant with our model. Then a series of iterative ergonomic evaluations with real users will produce new knowledge to be integrated in the rules, and possibly shed some light on the benefits of multimodal supervision in VR.

References

[All83] ALLEN J. F.: *Maintaining Knowledge about Temporal Intervals*, vol. 26 of *Communications of the ACM*. November 1983, ch. 11, pp. 832–843.

[BZ00] BARASS S., ZEHNER B.: Responsive sonification of well-logs. In *International Conference on Auditory Display* (Atlanta, Georgia, USA., 2000).

[FK94] FITCH W. T., KRAMER G.: *Sonifying the Body Electric:Superiority of an Auditory over a Visual Display in a Complex, Multivariate System*. G. Kramer, Addison-Wesley Publishing Company, 1994, ch. XVIII, pp. 307–326.

[Fri74] FRIEDES D.: Human information processing and sensory modality: Cross-modal functions, information complexity, memory and deficit. *Psychological Bulletin 81(5)* (1974), 284–310.

[GTB01] GOSLINE A. H., TURGAY E., BROUWER I.: Haptic illusions: What you feel isn't always what you get. In *Human Interface Technologies 2001* (2001), pp. 19–22.

[IBL*99] INFED F., BROWN S. W., LEE C. D., LAWRENCE D. A., DOUGHERTY A. M., PAO L. Y.: Combined visual/haptic rendering modes for scientific visualization. In *Proc. ASME Dynamic Systems and Control Division* (Nashville, TN, November 1999), DSC-Vol. 67, Int. Mech. Engr. Cong. & Expo., pp. 93–99.

[LCK*] LÉCUYER A., COQUILLART S., KHEDDAR A., RICHARD P., COIFFET P.: Pseudo-haptic feedback : Can isometric input devices simulate force feedback? IEEE Int. Conf. on Virtual Reality, pages 83-90, New Brunswick, US, 2000.

[MO97] MCLAUGHLIN J. P., ORENSTEIN B. J.: Haptic rendering of 3d seismic data. In *The Second PHANToM User's Group Workshop* (Cambridge, Massachusett, USA, MIT, 1997).

[NC93] NIGAY L., COUTAZ J.: A design space for multimodal systems - concurrent processing and data fusion. In *INTERCHI '93 - Conference on Human Factors in Computing Systems* (Amsterdam, 1993), Addison Wesley, pp. 172–178.

[RBVB06] ROUSSEAU C., BELLIK Y., VERNIER F., BAZALGETTE D.: A framework for the intelligent multimodal presentation of information. *Signal Processing Journal, Special issue on Multimodal Interfaces* (April 2006).

[RDLTH01] ROBLES-DE-LA-TORRE G., HAYWARD V.: Force can overcome object geometry in the perception of shape through active touch. *Nature 412* (2001), 445–448.

[SDL96] SALZMAN M. C., DEDE C., LOFTIN B.: Sciencespace: Virtual realities for learning complex and abstract scientific concepts. In *IEEE Virtual Reality Annual International Symposium* (1996), New York: IEEE Press, pp. 246–253.

[VN00] VERNIER F., NIGAY L.: A framework for the combination and characterization of output modalities. In *DSV-IS'00* (Limerick, Ireland, 2000), pp. 32–48.