Workshop in Virtual Reality Interactions and Physical Simulation VRIPHYS (2008)
F. Faure, M. Teschner (Editors)

From Interactive to Immersive Molecular Dynamics

N. Férey [1] and O. Delalande[1] and G. Grasseau[2] and M. Baaden[1]

[1]Institut de Biologie Physico-Chimique, Laboratoire de Biochimie Théorique, UPR CNRS 9080, France
[2]Institut du Développement et des Ressources en Informatique Scientifique, IDRIS, France

Abstract

Molecular Dynamics simulations are nowadays routinely used to complement experimental studies and overcome some of their limitations. In particular, current experimental techniques do not allow to directly observe the full dynamics of a macromolecule at atomic detail. Molecular simulation engines provide time-dependent atomic positions, velocities and system energies according to biophysical models. Many molecular simulation engines can now compute a molecular dynamics trajectory of interesting biological systems in interactive time. This progress has lead to a new approach called Interactive Molecular Dynamics. It allows the user to control and visualise a molecular simulation in progress. We have developed a generic library, called MDDriver, in order to facilitate the implementation of such interactive simulations. It allows one to easily create a network connection between a molecular user interface and a physically-based simulation. We use this library in order to study a biomolecular system, simulated by various interaction-enabled molecular engines and models. We use a classical molecular visualisation tool and a haptic device to control the dynamic behavior of the molecule. This approach provides encouraging results for interacting with a biomolecule and understanding its dynamics. Starting from this initial success, we decided to use Virtual Reality (VR) functionalities more intensively, by designing a VR framework dedicated to immersive and interactive molecular simulations. This framework is based on MDDriver, on the visualisation toolkit VTK, and on the vtkVRPN library, which encapsulates the VRPN library into VTK.

Categories and Subject Descriptors (according to ACM CCS): I.6.6 [Computing Methodologies]: Simulation and Modelling Simulation Output Analysis H.5.2 [Information Systems]: Information Interfaces and Presentation Graphic User Interface and Haptic I/O

1. Introduction

Molecular Dynamics (MD) is a simulation method that allows researchers to obtain complementary data with respect to experimental studies and to overcome some of their limitations. Current experimental techniques do not allow to observe the full dynamics of a macromolecule at atomic detail. In return, experiments provide the structures, i.e. the spatial atomic positions, for numerous biomolecular systems, which are used as starting point for simulation studies.

In order to predict, to explain and to understand experimental results, researchers have developed a variety of biomolecular representations and algorithms. They allow to simulate the dynamic behavior of macromolecules at different scales, ranging from detailed models using quantum mechanics or classical molecular mechanics to more approximate representations. These simulations are controlled *a priori* by complex and empirical settings. Most researchers visualise the result of their simulation once the computation is finished. Such post-simulation analysis often makes use of specific molecular user interfaces, by reading and visualising the molecular 3D configuration at each step of the simulation. This approach makes it difficult to interact with a simulation in progress. When a problem occurs, or when the researcher does not achieve to observe the predicted behavior, the simulation must be restarted with other settings or constraints. This can result in the waste of an important number of compute cycles, as some simulations last for a long time: several days to weeks may be required to reproduce a short timespan, a few nanoseconds, of molecular reality. Moreover, several biomolecular processes, like folding or large conformational changes of proteins, occur on even longer timescales that are inaccessible to current simulation techniques. It can thus be necessary to

impose empirical constraints in order to accelerate a MD simulation and to reproduce an experimental result. These constraints have to be defined *a priori*, rendering it difficult to explore all possibilities in order to examine various biological hypothesis.

A new approach allowing to address these problems has emerged recently: Interactive Molecular Dynamics (IMD). IMD consists in visualising and interacting with a simulation in progress, and provides user control over simulation settings in interactive time. A few Molecular User Interfaces (MUI), such as *VMD* [HDS96], offer such capabilities, providing methods to interface with a molecular simulation engine, for example *NAMD* [PBW*05]. In the next section we present a brief overview of the state-of-the-art.

1.1. An overview of Interactive Molecular Dynamics

Since the breakthrough initiated by the *Sculpt* precursor program [SRRBJ94], the interactive molecular simulations field has been developing continuously. Initial interactive experiments using molecular mechanics techniques gave quickly rise to "guided" dynamics simulations [WW99] [WW02] or *Steered Molecular Dynamics* (*SMD*) [IBG*01] [LPH96]. The interest for these methods increased with the enhancement of simulation accuracy and thanks to the exciting new possibilities for dynamic structural exploration of very large and complex biological systems. In the emerging Interactive Molecular Dynamics (IMD) approach, steering forces are applied interactively with a chosen amplitude, direction and application point. This enables the user to explore the simulation system while receiving instant feedback information from real-time visualisation or haptic devices [Rap97].

Recently, Schulten's group has carried out several convincing applications of IMD simulations to macromolecular structures [GTS03] [SGS01]. This effort lead to the design of two efficient software tools greatly facilitating the process of setting up an IMD simulation: *NAMD* and *VMD* ([PBW*05], [NHK*95]). Similar projects proposing an interactive display for molecular simulations exist, such as the *Java3D* interface proposed in [KM03] and [KM01], or the *Protein Interactive Theater* [PHM*99].

With fast generalization of new computer hardware devices and increasing accessibility to powerful computational infrastructures, IMD simulations show a fast and promising evolution, even for very large molecular systems (over 100.000 atoms). Such applications are now in the reach of state-of-the-art desktop computing. This evolution was possible given the strong increase in raw computing power leading to faster and bigger processing units (multi-processors, multi-core architectures). Currently ongoing technological developments such as GPU computing, PPU design and the spread of parallelizable entertainment devices (PS3, Cell) with specific graphic and processing capabilities open exciting new opportunities for interactive calculations. These approaches could provide even more processing power for highly parallelizable computational problems, for instance by differentiating the parallelisation of molecular calculations and graphical display functionalities. Given these developments, the range of accessible computational methods and representations is bound to grow. It may soon be possible to extend the IMD approach to first principle quantum mechanical (QM) or mixed QM/MM calculations. Such refined physical modelling methods increase the accuracy of the calculation. Alternatively, the precision of a given model can be improved by a better description of the molecular context and environment. This can be achieved via multi-scale simulations [BL07], which would largely benefit from interactive approaches leading to important advantages with respect to the study of complex biological systems. However, the raw increase in computer speed alone is not sufficient to grant a successful future evolution of the IMD approach. In addition, it is necessary to develop adapted software solutions dedicated to the IMD approach. This approach is generally more efficient [GTS03], as is commonly admitted in the numeric simulation field.

1.2. Objectives

The goal of our work is to extend the IMD approach to a broader range of simulation engines, as the use of a specific simulation sofware or model often depends on the studied biological system. We have developed a generic and independent library, called *MDDriver*, which allows us to easily interface molecular simulation engines with molecular visualisation tools through a network connection. As a first step, we have rendered the calculation modules easily interchangeable while keeping the existing *VMD* user interface as MUI. We have then examined alternative approaches for the visualisation and user interactions, which are also interchangeable via *MDDriver*. In order to take into account the real needs of a user, we have applied our approach to current research projects in molecular simulation of biological macromolecules. The *MDDriver* library enabled us to study the dynamic behavior of important biological systems, and the results of these studies convinced us that VR devices offer innovative functionalities such as interactive control during the simulation, allowing the user to better understand structural and mechanical biomolecular behavior. The main motivation in this work is to provide a VR framework allowing to quickly prototype new features and modules especially designed for the IMD approach in a VR context.

2. *MDDriver* to couple simulation and user interfaces

In the context of the *VMD/NAMD* approach, the IMD network protocol [SGS01] was developed in order to

interface the *VMD* MUI with the *NAMD* MD engine. However, the use of a specific simulation engine and MUI strongly depends on the studied biological system and on user habits. Adding IMD capabilities to other simulation engines and molecular models as well as to a variety of MUIs (in addition to *VMD* and *NAMD*) enables a whole range of new possibilities in interactive molecular simulations. This approach allows us to address a larger user community working on molecular modeling and simulations, sometimes based on their own home-made simulation engines. Following these motivations, we developed a generic and independent library, called *MDDriver*, inspired by the *VMD/NAMD* approach.

We have encapsulated the IMD protocol in the *MDDriver* library, allowing a developer to easily adapt MUI code and MD code in order to extend them with IMD features. This interface provides functions for the exchange of specific data structures over a network. These are atom positions and system energies, computed for each simulation step by the MD engine (server part), and user-applied forces on a selected atom set (see Figure 1; client part). We also extended the IMD protocol in *MDDriver* in order to be able to work on the DEISA network (Distributed European Infrastructure for Supercomputing Applications).

This approach was tested, applied and improved by integrating calls to the *MDDriver* library into the *GROMACS* simulation engine [HKVL08], thus rendering the simulation interactive via a MUI. We have used *VMD* as MUI in order to study the molecular behavior of the Guanylate Kinase Enzyme (GK) and SNARE systems using all-atom and coarse-grained modeling methods [BL07] with *GROMACS*. In the coarse grain model, heavy atoms are grouped into pseudo atoms in order to obtain a low resolution representation and decrease the number of particles in the system. In this way, it is possible to substantially increase the timescale of the molecular simulation.

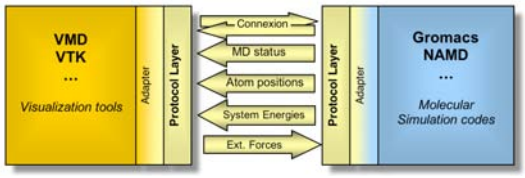

Figure 1: *MDDriver library for interfacing a Molecular Dynamics simulation with a Molecular User Interface*

3. From Interactive to Immersive Molecular Dynamics

The *MDDriver* library was a necessary first step before developing a VR application dedicated to IMD. It allowed us to simplify the MD simulation code and MUI coupling. The *MDDriver* library enabled us to study the dynamic behavior of important biological systems, such as the GK or the SNARE protein complex, simulated by different calculation engines and represented by models at varying resolution. As a natural extension to this work, we started to intensify the use of VR functionalities in a new interactive application for molecular dynamics analysis. The main motivation was to provide a VR framework allowing to quickly prototype new features and modules especially designed for the IMD approach in a VR context. Such applications would be difficult to realise with an end-user oriented package such as *VMD* because of its complex software architecture and its desktop oriented approach. We have based our design on three developments: the *MDDriver* library, the visualisation toolkit *VTK*, and the *vtkVRPN* library. *VTK* provides the ability to quickly develop new visual renderings and user interactions; *vtkVRPN* is a library to properly integrate VR devices into *VTK* through *VRPN*.

3.1. Molecular visualisation with *VTK*

The open source project *VTK* is supported by a large user community. We chose this toolkit because of its high-level functionalities for scientific vizualisation, and of its large user base. The main motivation for this approach is to provide a VR framework allowing to rapidly prototype new features and modules especially designed for controlling molecular simulation engines in interactive time. *VTK* provides bindings to script languages such as *Python* and *TCL*, commonly used by biostructural researchers. Finally, *VTK* makes it easy to implement standard molecular visuals such as atom, "ball and stick", ribbon, or cartoon representations.

In Figure 2 we show how to use a *VTK* pipeline in order to visualise and interact with a dynamic "ball and stick" representation. Each sphere represents an atom (ball) and each cylinder a chemical bond between two atoms (stick; see Figure 4). The *vtkPDBReader* class reads the initial molecular conformation (atom types, atom positions, and chemical bonds between atoms) that was obtained by experimental studies and stored in the PDB molecular file format. It is then transformed into a *vtkPolyData* instance, designed in *VTK* for describing polyline or polymesh data (points and topology). On the one hand, a *vtkGlyph3D* instance filters points (atom positions) of this *vtkPolyData* instance considering each one as a sphere, using *vtkSphereSource*. It deals with the color and radius mapping according to atom type. On the other hand, a *vtkTubeFilter* instance filters lines (chemical bonds) considering each one as a cylinder. The *vtkPolyDataMapper* translates these results into *OpenGL* commands. Other commonly used molecular representations, such as ribbons or secondary structures, and less used ones, like volume rendering and isosurface representations used to illustrate electrostatic potential around the molecule, can also be implemented using native *VTK* classes.

The dynamic behavior of this representation is computed

by a molecular simulation engine, interfaced via the *VTK* client using the *MDDriver* library. Dynamic events trigger the position update of all atoms. Timer events trigger the geometry update and the graphical rendering.

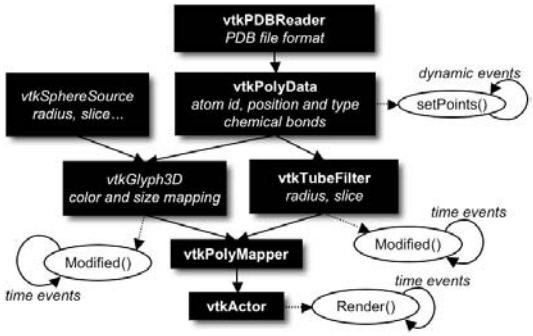

Figure 2: *Dynamic "ball and stick" representation using the VTK pipeline*

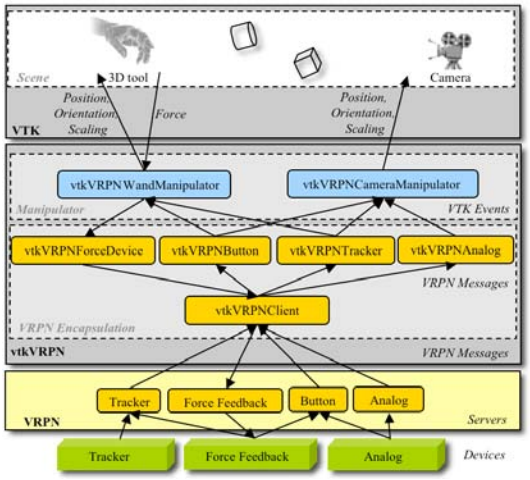

Figure 3: *VRPN encapsulation into VTK through the vtkVRPN library*

3.2. Integrating VR devices in *VTK* with *vtkVRPN*

The *VTK* toolkit offers stereo rendering, but does not provide other native VR functionalities, such as tracker or haptic device management. We can cite an approach [KvL07] extending *VTK* to a VR context in a platform-independent and sofware-independent way. Our approach is based on *VRPN* [THS*01], a library managing many different VR devices. *VRPN* is used to handle actual user hardware and software settings related to VR device management. *VRPN* is probably the most widely spread VR framework among biostructural researchers. It is also used by the *VMD/NAMD* tools. The *vtkVRPN* library [JG05] was developed to interface *VRPN* and *VTK* in order to enable the efficient use of VR devices within *VTK*. This allowed us to easily integrate all the devices supported by *VRPN* with little change to any existing *VTK* application. The *vtkVRPN* library is composed of two layers. The first one, the *VTK Encapsulation layer*, encapsulates *VRPN* into *VTK* converting *VRPN* messages into VTK events. The second layer, the *Manipulator layer*, is composed of classes reacting to *VTK* events. The Manipulator layer allows to interact with graphical objects such as a 3D cursor or a hand avatar (*vtkWandManipulator*) as well as with the camera (*vtkCameraManipulator*) in the *VTK* scene. The class *vtkWandManipulator* also manages haptic feedback, using high level *VTK* functionalities for simple feedback, and native *VRPN* capabilities for computationally demanding feedback.

3.3. Interactive paradigm for driving a simulation

The haptic device is used to control the direction of the forces applied to selected atoms. Moreover the haptic feedback offers the possibility to adjust the amplitude of the force, and to take a user defined force scale factor into account. The interaction paradigm to interactively impose forces on particles contains two stages. The first stage is the selection of a single particle or a set of particles using a 3D tool attached to a haptic device using its buttons. In a second stage, we use the force model described in [SGS01] in order to compute the forces applied to the selected atoms. The main idea of this paradigm is to link the selected atoms and the 3D haptic tool with a spring. Instead of providing haptic rendering of forces or positions computed in the simulation, the force feedback only depends on the spring length. The user's force perception is governed by a kind of action/reaction principle. The more the selected objects are slow to react to applied forces in the simulation, the more the rendered force seems to be strong. Indeed in this case the spring length decreases slowly. Therefore the user can feel the deformability of the selected sub-system due to external factors. Another advantage of this paradigm is that the low physical simulation framerate (5 to 10 Hz) does not cause instabilities and does not affect the haptic feedback, because it only depends on the spring length computed in haptic time (300 to 1000Hz). This force feedback interaction with the molecular scene was implemented in our framework as in the *VMD* tool [PBW*05]. We improved our implementation by adding haptic and visual feedback during the selection stage, unavailable in VMD. The resulting forces are rendered by haptic feedback if a haptic device is used, or by visual feedback such as the blue arrows shown in the top left part of Figure 4. Simultaneously, forces are sent to the simulation engine using the *MDDriver* library. A new aspect with respect to the haptic feedback concerns the user's perception of forces in a coarse-grain simulation. Indeed the force scale factors have to be empirically chosen according to the

resolution of the simulation. We should notice that it may be necessary to adjust this parameter during an IMD simulation in order to obtain a better force efficiency (depending on the nature of the system studied). This is a necessary but quite delicate task. In the case of an overestimation, one may induce non pertinent structural distortions. Furthermore force scaling seems to require a precise and judicious atom selection in order to prevent irreversible or damageable structure integrity losses.

4. Applications

In the near future, IMD developments will probably allow to further increase the attractivity of the method by enabling more and more challenging applications. These could reach from the implementation of instantaneous data analysis, free energy computations, docking facilities or the simulation of non-reversible processes up to interactively tractable chemical reaction processes. In this perspective, we have explored combinations of all-atom or coarse-grained modeling methods implemented in the *GROMACS* software package within IMD simulations driven via the *VMD* tool ([HDS96]). We have tested this approach on a large selection of complex biological systems using our *MDDriver* library that was developed as a generic platform combining both a visualisation interface and a user's preferred calculation code, standardizing and optimizing the particle coordinate exchange. This section presents some of these biomolecular systems studied using the *MDDriver* framework and/or VR framework that was described in the previous sections. In these studies, we use a 3DOF tracker or an analog device allowing us to control the camera, and haptic devices (SensAble© PHANTOM® Premium 1.5 and Omni®) for driving molecular simulations.

4.1. *MDDriver* for coupling *GROMACS* and *VTK*

The purpose of the first study was to validate concepts of our IMD approach on a simple test system, and to demonstrate that the coupling of *GROMACS* to an alternative MUI, other than VMD, is easy to achieve using *MDDriver*. We have used our *VTK* VR application as MUI. The basic system is a polypeptide of a few hundred atoms (see Figure 4).

While it was straight-forward to validate the application, we observed some performance issues for interactive rendering with *VTK*. Despite the fact that *VTK* has a well-adapted visualisation model for static molecular visualisation and interaction with the scene via high-level functionalities, a dynamic scene imposes additional constraints. We are now working on increased rendering performance, compensating for the dynamic aspect of IMD which is currently limiting the interactivity to several thousand atoms.

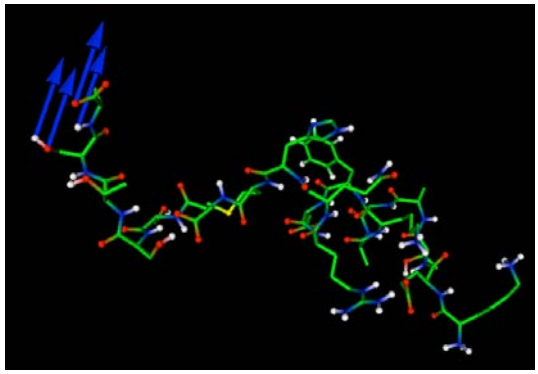

Figure 4: *Haptic control of a polypeptide simulation in "ball and stick" representation*

4.2. *MDDriver* for coupling *GROMACS* and *VMD*

In the next two studies, we have examined more complex biomolecular systems: the GK enzyme and the SNARE complex. Structures for these molecules are provided by experimental methods such as Nuclear Magnetic Resonance or X Ray cristallography. In the two next case studies we use both all-atom and coarse grain models for studying the dynamic behavior of these molecules at different timescales. We recall that in the coarse grain model, atoms are grouped into pseudo atoms, allowing to decrease the number of particles in the system and therefore to increase the timescale of the molecular simulation.

The GK enzyme has a U shape and can adopt either a closed or an open conformation (see Figure 5). The closure mechanism of GK consists in increasing the proximity of two substrate binding sites, for GMP and ATP, both essential for the enzymatic reaction. The goal of our study is to understand which parts of this system are involved in the closure mechanism. This mechanism has been investigated at two levels of detail. The first level corresponds to the detailed all-atom model, the second to a lower resolution coarse-grain model. Prospective tests using coarse-grain simulations allowed for the efficient exploration of a broad range of possibilities to close the enzyme, trying to reach a closed conformation similar to the available experimental structures. Figure 5 shows a ball and stick representation of the protein, considering specific architectural units relevant for investigating the closure mechanism (yellow and orange areas). Figure 6 (left) shows a coarse-grain representation of the GK complex. The crucial role of one loop (highlighted in yellow in Figure 5) in the initiation of GK's closure could thus be identified. It was then confirmed in a second phase using more detailed all-atom simulations. Understanding the features of this early intermediate state occurring as an impulse for the closure mechanism allows us to propose a novel mechanistic hypothesis. The loop move could be initiated by GMP docking, which may drive this loop via

long range electrostatic interactions. When the loop draws closer to the other side of the enzyme, conformational changes could be triggered, subsequently inducing a global closure of the enzyme. The interactive exploration of the simulation using the haptic modality lead us to this innovative theoretical hypothesis.

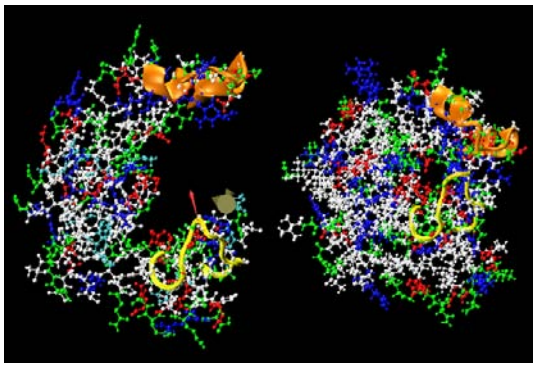

Figure 5: *GK in the open state (left) and the closed state (right) with haptic control of the yellow area (red arrow).*

Concerning the SNARE complex, we assume that the membrane protein anchoring and the assembly of the four helical bundle is linked to specific mechanical properties. The IMD simulations carried out for both the coarse-grain and all-atom SNARE system have generated theoretical data helping us to improve the coarse-grain method itself (still under development), and to gain insight into the dynamic structural properties of this biological system. The trans-membrane insertion of the syntaxin and synaptobrevin (in red and blue in Figure 6) helices into a lipid bilayer has been extensively studied on a reduced SNARE system. We essentially focus on the reorientation and repositioning of specific residues inside the membrane (or at its surface) after a perturbation carried out by the user. This helps to compare the relaxation of the simulated system to experimental measurements. The force feedback obtained via haptics upon pulling a lipid molecule out of the membrane is very informative with respect to parameterization. Concerning a larger, more extended simulation model, a simple pull on the terminal part of the four-helix bundle (in yellow, gray, orange and green in the right part of Figure 6) initiates an important perturbation of the membrane. A stretching of the global helical bundle is also noticed. We have further observed the structural and mechanical response of the whole helix bundle to an imposed force. This revealed a different behaviour of the syntaxin and synaptobrevin trans-membrane helices concerning their anchoring strength into the lipid bilayer. The IMD simulation hinted at a higher stability for the synaptobrevin trans-membrane helix compared to the syntaxin one. Finally, IMD simulations allowed us to examine the structural integrity of the SNARE protein assembly. One possible test concerns the self-association of the four SNARE helices guided by their hydrophobic complementarity. It can be assessed by trying to unzip the helix bundle and letting it relax back to system equilibrium.

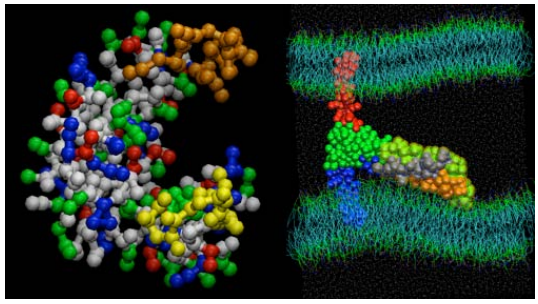

Figure 6: *Using haptics for inducing large conformational changes (left, GK system) and exploring protein mechanical properties (right, SNARE complex in a double lipid bilayer).*

We will briefly comment on the performance obtained for the *MDDriver* library implementation into the *GROMACS* MD code. The data (coordinates, status and forces) transfer rate between calculation and visualization modules essentially depends on the size of the simulated system. The force application component slightly affects IMD performance in particular for small systems, depending essentially on the selected to total particle ratio (increasing data exchange). The display rendering step can be a major limitation for very large systems, although all measurements were done with the lightest representation of the scene for convenient IMD use in VMD. We should mention two remaining issues concerning latencies in the scene display with our desktop installation. One is related to the pick/unpick event of the haptic device and deals with a reactivity problem that is well-known in a VRPN server use context. Another latency display problem occurs without interaction with the molecular scene and is a direct effect of the chosen MD calculation protocol. The IMD rate is slowed down by the re-computation of the atom neighbour list as needed by *GROMACS* (more than 50% exchange rate loss every 10 simulation steps in our case) for the computation of the pairwise interactions between atoms. In the post-simulation rendering of a classical trajectory these latencies in the interactive molecular scene display are innocuous. Obviously they have drastic effects on an IMD simulation, temporarily freezing the interactive session and perturbing the haptic rendering.

In the context of deploying *GROMACS* IMD simulations via *MDDriver* on a large computing infrastructure, a similar performance has been achieved. This confirms the robustness of the *MDDriver* library coupled to parallelized applications. The performance of the display and interaction part is the main limitation for IMD simulations of large molecular systems.

4.3. Limitations

Some unsolved intrinsic problems remain with respect to the IMD approach. Their origin currently lies in the limited simulation timescale and the necessity to apply high forces in order to induce significant conformational changes. This limit is dependent on the computational method used. In all IMD studies published up to date, all-atom molecular dynamics simulations were employed. The short integration timestep of these simulations limits the application range of the IMD technique. In order to effectively observe large structural changes, such classical non-interactive MD simulations are now performed for hundreds of nanoseconds or more. However, corresponding IMD simulations are intrinsically restricted to the time the user can stay in front of a computer as well as the user's human timescale for interaction that lies between seconds and minutes. In this study, we present the application of the IMD approach to simplified modelling methods, such as coarse-graining ([BHI*07], [BS06], [NLSK04], [MdVM04]), allowing for larger timesteps. In our opinion, these techniques are more suitable for interactive experiments and represent a better compromise for interactive computations whilst enabling exciting applications helping to solve biological problems.

5. Discussion

The work presented in this paper is the first milestone in a larger project currently in progress. Here we present a VR framework architecture, including the *MDDriver* library. It was designed and tested in order to easily interact with various simulation engines such as *NAMD* or *GROMACS*, a *VTK*-based MUI and *vtkVRPN* for integrating VR devices into *VTK*. The *MDDriver* library offers a simple, modular and generic solution to combine any coordinates-based calculation code with various visualization programs. By this means the IMD simulation approach, a powerful tool for the exploration of biomolecular structures in large biological systems, is now easily accessible to desktop or VR computational environments. We insist on the fact that the *MDDriver* library was designed for easy integration into any molecular simulation engine providing time series of particle positions. Indeed there are many approaches capable of simulating the dynamic behavior of biomolecules, such as lattice simulations, elastic networks, coarse grain models or even quantum mechanical and semi-empirical methods. We show through several studies that this framework is able to interact with a variety of MUIs, such as *VMD* or our own interactive MUI based on *VTK* and *vtkVRPN*. Many application studies such as the investigation of the dynamic behavior of the Guanylate Kinase enzyme or the SNARE complex have also provided encouraging feedback on the potential of this approach.

The use of a simple haptic paradigm allowed us to interact with different molecular simulation engines varying the level of detail of the molecular model (coarse grain and all-atom) and therefore accessing different timescales. This investigation lead us to a new hypothesis on GK's closure mechanism and allowed us to study the complex dynamic behavior of the SNARE system. Moreover, simulating and interacting with a molecular system based on different modelling scales allows researchers to collect cooperative data and provides information about accuracy, pertinence or implementation of simulation methods in terms of reliability of energies and physical or chemical properties. Finally, the IMD approach appears as a powerful tool to improve new simulation methods under development or assess the impact of simulation parameters.

It has to be noted that the time scale of a simulation, particularly with respect to VR investigations in interactive time, imposes certain limits. In order to produce an effective event on a dynamic molecular structure, it may be necessary to impose an important force beyond the range of pertinent biophysical energies. This problem needs to be addressed in future work. Another bottleneck concerns performance. While *VTK* has a well-adapted visualisation model for static molecular visualisation and interaction with the scene via high-level functionalities, a moving scene imposes additional constraints. We have observed weak performance using *VTK* on a system of 2500 atoms with the framerate dropping below 10 frames per second. With several hundreds of atoms the framerate was about 24 frames per second. This issue can be attributed to *VTK*, because the computing overhead using *MDDriver* with *GROMACS* against *GROMACS* alone is less than 6 percent. These limitations can probably be overcome by separating the simulation system into sub-parts. One sub-part would concern about 80% of all atoms, including solvent, surrounding the studied biomolecule. These particles are important for a pertinent biomolecular simulation with respect to rendering, but not for the interaction. We are working on the optimization of *VTK*, using a classical graphical *VTK* representation for atoms of the sub-part concerned by the interaction, and more powerful techniques like shaders for the solvent atoms, in order to be able to handle and interact with bigger molecular systems. Our objective is to deal with at least several hundred thousand atoms.

Another interesting aspect of this work is the way user interaction is mediated to the simulation engine. We currently use forces, as they are a natural way to impose external constraints on an MD simulation. However, sending forces to the simulation engine might not be the only paradigm for interacting with the simulation. Other modalities such as imposing changing atomic coordinates can be considered. This approach should be straightforward for applications such as interactive molecular rigid body docking, where the goal is to study the dynamic assembly and interaction of two rigid proteins, by applying global modifications to the atomic positions of the selected molecule. We are currently investigating this approach in our

in-house docking software. We encourage experimenting with our framework which will soon be available at http://mddriver.sourceforge.net with movies and tutorials.

6. Acknowledgements

This work was supported by the DEISA Consortium (co-funded by European Union, FP6 508830/031513) and IDRIS (CNRS's National Supercomputer Center). This work is also supported by two ANR grants for software development (ANR-07-CIS7-003-01) and biological applications (ANR-06-PCVI-0025). We like to thank J.P. Nominé, G. Journé, and C. Guilbaud for providing the *vtkVRPN* library, which was designed at the "Commissariat à l'Energie Atomique".

References

[BHI*07] BOND P. J., HOLYOAKE J., IVETAC A., KHALID S., SANSOM M. S. P.: Coarse-grained molecular dynamics simulations of membrane proteins and peptides. *J. Struct. Biol. 157*, 3 (2007), 593–605.

[BL07] BAADEN M., LAVERY R.: There's plenty of room in the middle: Multiscale Modelling of Biological Systems. *Recent Advances in Structural Bioinformatics* (2007), 173–196.

[BS06] BOND P. J., SANSOM M. S.: Insertion and assembly of membrane proteins via simulation. *J. Am. Chem. Soc. 128*, 8 (2006), 2697–2704.

[GTS03] GRAYSON P., TAJKHORSHID E., SCHULTEN K.: Mechanisms of selectivity in channels and enzymes studied with interactive molecular dynamics. *Biophys. J. 85* (2003), 36–48.

[HDS96] HUMPHREY W., DALKE A., SCHULTEN K.: VMD - Visual Molecular Dynamics. *J. Mol. Graph. 14* (1996), 33–38.

[HKVL08] HESS B., KUTZNER C., VANDERSPOEL D., LINDAHL E.: GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory and Comp. 4*, 3 (2008), 435–447.

[IBG*01] ISRALEWITZ B., BAUDRY J., GULLINGSRUD J., KOSZTIN D., SCHULTEN K.: Steered molecular dynamics investigations of protein function. *J. Mol. Graph. 19* (2001), 13–25.

[JG05] JOURNÉ G., GUILBAUD C.: *Intégration de périphériques de réalite virtuelle dans des applications de visualisation scientifique au sein la plateforme vtkVRPN*. CEA-R-6092 Research Note, 2005.

[KM01] KNOLL P., MIRZAEI S.: Quick and easy interactive molecular dynamics using Java3D. *J. Comp. Sci. Eng. 3* (2001), 98–104.

[KM03] KNOLL P., MIRZAEI S.: Development of an interactive molecular dynamics simulation software package. *Rev. Sci. Instrum. 74* (2003), 2483–2487.

[KvL07] KOK A. J., VAN LIERE R.: A multimodal virtual reality interface for 3D interaction with VTK. *Proc. of Knowledge and Information Systems 13*, 2 (2007), 197–219.

[LPH96] LEECH J., PRINS J. F., HERMANS J.: SMD : visual steering of molecular dynamics for protein design. *J. Comp. Sci. Eng. 3* (1996), 38–45.

[MdVM04] MARRINK S. J., DE VRIES A. H., MARK A. E.: Coarse grained model for semiquantitative lipid simulations. *J. Phys. Chem. B 108*, 2 (2004), 750–760.

[NHK*95] NELSON M., HUMPHREY W., KUFRIN R., GURSOY A., DALKE A., KALE L., SKEEL R., SCHULTEN K.: MDscope - a visual computing environment for structural biology. *J. Comp. Phys. Comm. 91* (1995), 111–133.

[NLSK04] NIELSEN S. O., LOPEZ C. F., SRINIVAS G., KLEIN M. L.: Coarse grain models and the computer simulation of soft materials. *J. Phys. Condens. Matter 16* (2004), 481–512.

[PBW*05] PHILLIPS J. C., BRAUN R., WANG W., GUMBART J., TAJKHORSHID E., VILLA E., CHIPOT C., SKEEL R. D., KALE L., SCHULTEN K.: Scalable Molecular Dynamics with NAMD. *J. Comp. Chem. 26* (2005), 1781–1802.

[PHM*99] PRINS J. F., HERMANS J., MANN G., NYLAND L. S., SIMONS M.: A virtual environment for steered molecular dynamics. *Fut. Gen. Comp. Sys. 15* (1999), 485–495.

[Rap97] RAPAPORT D. C.: Interactive Molecular Dynamics. *Physica A. 240* (1997), 246–254.

[SGS01] STONE J. E., GULLINGSRUD J., SCHULTEN K.: A system for interactive molecular dynamics simulation. *Proc. of Interactive 3D Graphics* (2001), 191–194.

[SRRBJ94] SURLES M. C., RICHARDSON J. S., RICHARDSON D. C., BROOKS-JR. F. P.: Sculpting proteins interactively : Continual energy minimization embedded in a graphical modelling system. *Protein Science 3* (1994), 198–210.

[THS*01] TAYLOR II R. M., HUDSON T. C., SEEGER A., WEBER H., JULIANO J., HELSER A. T.: VRPN: A Device-Independent, Network-Transparent VR Peripheral System. *Proc. of Virtual Reality Software and Technology* (2001), 55–61.

[WW99] WU X., WANG S.: Enhancing systematic motion in molecular dynamics simulation. *J. Chem. Phys. 110* (1999), 9401–9410.

[WW02] WU X., WANG S.: Direct observation of the folding and unfolding of b-hairpin in explicit water through computer simulation. *J. Am. Chem. Soc. 124* (2002), 5282–5283.