

Chapitre 2

SHADDOCK – Système collaboratif pour la manipulation de molécules

Sommaire

2.1	Plates-formes collaboratives existantes	5
2.1.1	Architecture du système	6
2.1.2	Vue partagée	7
	Références	8
2.2	Simulation moléculaire en temps-réel	9
2.2.1	Les besoins	9
2.2.2	Les outils existants	10
	Références	13
2.3	Interaction avec la simulation moléculaire	14
2.3.1	Les besoins	14
2.3.2	Interaction par interface haptique	15
	Références	16
2.4	Nouveaux outils pour l'interaction	17
2.4.1	Amélioration de la sélection	17
2.4.2	Déformation par groupe d'atomes	17
2.4.3	Outil de désignation et attraction	18

2.1 Plates-formes collaboratives existantes

SHADDOCK est le nom de l'Environnement Virtuel Collaboratif (EVC) développé dans le cadre de cette thèse. Les EVCs existants sont nombreux mais aucun ne

convient à nos besoins. Plusieurs choix d’architectures ont été nécessaires et sont exposés dans cette section. Une vue schématique de la plate-forme SHADDOCK est détaillée sur la figure 2.1. Les différents éléments de ce schéma seront expliqués dans les sections qui vont suivre. La section 2.1.1 tranchera sur la question de l’architecture pair-à-pair comparée à l’architecture client/serveur. Ensuite, la gestion d’une vue partagée est abordée dans la section 2.1.2 page ci-contre.

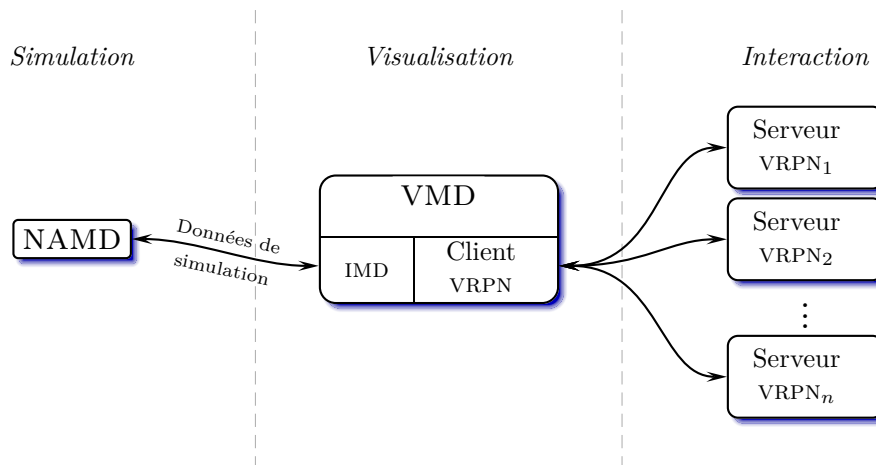


Figure 2.1 – Schéma de la plate-forme SHADDOCK

2.1.1 Architecture du système

Deux types d’architectures sont possibles pour les EVCS : client/serveur ou pair-à-pair. Parmi les architectures pair-à-pair, IGLESIAS et al. [2008] propose une tâche d’assemblage collaboratif assisté par l’haptique. J. KIM et al. [2004] étudie le déplacement collaboratif d’une boîte également assisté par l’haptique. Aucune des deux plate-formes proposées ne souffre d’instabilités notables dans le rendu haptique. Dans les EVCS, l’architecture pair-à-pair permet un rendu haptique relativement stable puisque chaque nœud du système gère sa propre simulation et n’est pas lié au latence voire au coupure du réseau.

Cependant, des travaux proposent également des architectures client/serveur avec des interactions haptiques. P. HUANG et al. [2010] propose une interaction haptique client/serveur pour la manipulation d’un jeu de construction par blocs. NORMAN et HAMZA-LUP [2010] s’intéresse particulièrement aux influences du réseau sur les interactions visuo-haptiques. L’avantage d’une architecture client/serveur est la cohérence de la simulation entre les différents nœuds du système. En effet, seul le serveur effectue la simulation et ensuite distribue les données de simulation aux différents nœuds.

MARSH et al. [2006] propose une comparaison de ces deux types d’architectures et en vient à la conclusion que l’architecture pair-à-pair est la plus performante

en terme de latence. En effet, l'architecture pair-à-pair permet un lien direct entre tous les nœuds du système. Dans le cas d'une architecture client/serveur, le serveur sert toujours d'intermédiaire ce qui augmente le nombre de rebonds des paquets réseaux.

Cependant, les systèmes pair-à-pair sont adaptés pour les environnements virtuels statiques ou faiblement dynamiques. En effet, des latences ou des coupures du réseau introduisent des erreurs de synchronisation entre les simulations ayant pour conséquence une divergence des résultats de simulation entre les différents nœuds. L'architecture client/serveur, bien que moins performante en terme de latence, permettra de conserver une cohérence de la simulation. La simulation est effectuée sur un serveur et les données de simulation sont transmises à tous les nœuds de l'EVC. Ce type d'architecture est très adapté pour les environnements virtuels dynamiques.

Étant donné les besoins en simulation, la plate-forme SHADDOCK est une architecture client/serveur.

2.1.2 Vue partagée

Dans un EVC, un point important est la disponibilité de la vue. Elle peut être soit privée pour chaque utilisateur, soit partagée par tous les utilisateurs.

L'un des premiers travaux sur la collaboration haptique est fourni par BASDOGAN et al. [2000] dans lequel deux utilisateurs placés dans deux pièces différentes doivent effectuer une tâche nécessitant de la synchronisation. OLIVEIRA et al. [2002] propose un EVC intégrant chaque utilisateur comme un avatar dans l'environnement virtuel. Dans les deux cas, chaque utilisateur n'a conscience des autres utilisateurs que par l'intermédiaire de l'EVC. La communication est alors restreinte aux modalités autorisées (strictement haptique et partiellement visuelle dans ces deux cas). L'impossibilité de communiquer oralement peut devenir un frein dans la collaboration.

Être conscient des faits et gestes des autres utilisateurs dans un EVC a prouvé son efficacité. SALLNÄS [2010] a beaucoup travaillé sur l'apport de l'haptique sur la conscience périphérique. Elle montre à plusieurs reprises l'intérêt d'être conscients des faits et gestes des autres utilisateurs ainsi que de l'état de l'environnement. CASERA et al. [2006] privilégie une vue publique à tous les utilisateurs pour améliorer la conscience périphérique. Enfin, TANG et al. [2006] a exploré la collaboration sur une table tactile : tous les utilisateurs évoluent autour de la même table tactile et possède donc une conscience accrue des faits et gestes de chacun.

SHADDOCK doit permettre la collaboration sur des tâches complexes. L'objectif est d'obtenir une collaboration la plus complète possible afin d'exploiter au maximum le potentiel de chaque utilisateur présent. C'est pourquoi SHADDOCK propose une vue publique permettant à tous les utilisateurs de se trouver au même endroit pour effectuer les tâches demandées. Ils ont ainsi la possibilité

de communiquer librement. La vue publique est assurée par un vidéoprojecteur projetant la vue sur un grand écran en face des utilisateurs. Les utilisateurs sont disposés de front face à l'écran.

Références

- BASDOGAN, Cagatay, Chih-Hao HO, Mandayam A. SRINIVASAN et Mel SLATER (déc. 2000). « An experimental study on the role of touch in shared virtual environments ». Dans *ACM Transaction on Computer-Human Interaction* 7.4, pages 443–460 (cf. page 7).
- CASERA, Steve, Hans-Heinrich NÄGELI et Peter KROPF (août 2006). « Improving usability of collaborative scientific visualization systems ». Dans *Proceedings of Visualization, Imaging, and Image Processing*. Tome 541, pages 180–186 (cf. page 7).
- HUANG, Pingguo, Yutaka ISHIBASHI, Norishige FUKUSHIMA et Shinji SUGAWARA (oct. 2010). « Interactivity improvement of group synchronization control in collaborative haptic play with building blocks ». Dans *Proceedings of the 9th Annual Workshop on Network and Systems Support for Games*. NetGames '10 2. Piscataway, NJ, USA : IEEE Press, pages 1–6 (cf. page 6).
- IGLESIAS, Rosa, Sara CASADO, Teresa GUTIÉRREZ, Alejandro GARCÍA-ALONSO, Wai YU et Alan MARSHALL (jan. 2008). « Simultaneous remote haptic collaboration for assembling tasks ». Dans *Multimedia Systems*. Tome 13. 4. Springer, Heidelberg, Germany, pages 263–274 (cf. page 6).
- KIM, Jung, Hyun KIM, Boon K. TAY, Manivannan MUNIYANDI, Mandayam A. SRINIVASAN, Joel JORDAN, Jesper MORTENSEN, Manuel OLIVEIRA et Mel SLATER (juin 2004). « Transatlantic touch : a study of haptic collaboration over long distance ». Dans *Presence : Teleoperators and Virtual Environments* 13.3, pages 328–337 (cf. page 6).
- MARSH, James, Mashhuda GLENCROSS, Steve PETTIFER et Roger HUBBOLD (mai 2006). « A network architecture supporting consistent rich behavior in collaborative interactive applications ». Dans *IEEE Transactions on visualization and computer graphics* 12.3, pages 405–416 (cf. page 7).
- NORMAN, Jonathan et Felix G. HAMZA-LUP (avr. 2010). « Challenges in the deployment of visuo-haptic virtual environments on the internet ». Dans *Proceedings of the 2010 Second International Conference on Computer and Network Technology*. Washington, DC, USA : IEEE Computer Society, pages 33–37 (cf. page 6).

OLIVEIRA, Jauvane C. de, XiaoJun SHEN et Nicolas D. GEORGANAS (nov. 2002). « Collaborative virtual environment for industrial training and e-commerce ». Dans *Virtual reality technologies for future telecommunications systems*. Sous la direction d'Algirdas PAKSTAS et Ryoichi KOMIYA. San Francisco, CA, USA : Willey, pages 79–102 (cf. page 7).

SALLNÄS, Eva-Lotta (juil. 2010). « Haptic feedback increases perceived social presence ». Dans *Proceedings of the 2010 international conference on Haptics - generating and perceiving tangible sensations : Part II*. Berlin, Heidelberg : Springer-Verlag, pages 178–185 (cf. page 7).

TANG, Anthony, Melanie TORY, Barry PO, Petra NEUMANN et Sheelagh CARPENDALE (avr. 2006). « Collaborative coupling over tabletop displays ». Dans *Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA : ACM, pages 1181–1190 (cf. page 7).

2.2 Simulation moléculaire en temps-réel

La plate-forme SHADDOCK permet de visualiser des simulations moléculaires en temps-réel. Dans cette section, nous commençons par identifier les besoins en terme de simulation dans la section 2.2.1 puis par exposer les solutions logicielles retenues dans la section 2.2.2 page suivante.

2.2.1 Les besoins

SHADDOCK permet d'effectuer la visualisation des molécules. La visualisation est un processus complexe qui nécessite des rendus variés et complets. En effet, devant le nombre important d'informations disponibles concernant une molécule, il est primordial d'avoir un rendu graphique de molécules clair et complet sans être surchargé. Cette tâche est effectuée par le logiciel VMD (voir section 2.2.2 page suivante).

Ensuite, SHADDOCK simule un environnement moléculaire. Un logiciel de simulation est nécessaire pour réaliser cette tâche. Il faut que ce logiciel puisse interagir avec VMD. De plus, il est nécessaire de pouvoir paramétrer finement la simulation (par exemple, fixer certains atomes). NAMD (*Scalable Molecular Dynamics*) est le parfait candidat (voir section 2.2.2 page 12).

Cependant, NAMD n'est pas conçu pour effectuer des simulations en temps-réel. Pour proposer une interaction aux utilisateurs, il est nécessaire d'avoir accès à une simulation interactive et en temps-réel. La solution à ce problème vient avec IMD (*Interactive Molecular Dynamics*) développé par l'*Institut für Theoretische und Angewandte Physik* (ITAP) et présenté en section 2.2.2 page 13.

2.2.2 Les outils existants

Trois outils permettent de fournir une simulation en temps-réel de molécules : **VMD** pour la visualisation, **NAMD** pour la simulation et **IMD** pour la simulation en temps-réel. Ces trois outils sont présentés dans les sections suivantes.

VMD (*Visual Molecular Dynamics*)

Les outils de visualisation moléculaire disponibles sont relativement nombreux. Parmi les plus populaires, on peut citer PyMOL [SCHRÖDINGER 2010], VMD [HUMPHREY et al. 1996], CHIMERA [PETTERSEN et al. 2004], RASMOL [SAYLE et MILNER-WHITE 1995] sans compter les nombreux dérivés permettant un affichage en ligne tel que Jmol [SOURCEFORGE.NET 2006] pour ne citer que le plus connu. PyMOL et VMD se distinguent particulièrement par leurs nombreuses fonctionnalités et leur large utilisation dans le milieu spécialisé.

PyMOL est probablement le logiciel de visualisation le plus utilisé par les experts du domaine car c'est le plus complet pour fournir des rendus graphiques de molécules très complets. Cependant, PyMOL ne permet pas l'affichage de simulations temps-réel ni la manipulation interactive de molécules.

VMD possède également une large gamme de rendus graphiques. Contrairement à PyMOL, VMD est adapté pour le rendu graphique en temps-réel de données de simulation. Il permet également la manipulation interactive de molécules. Les fonctionnalités de VMD sont nombreuses et seulement certaines ont été utilisées dans le cadre des expérimentations qui vont suivre. Elles sont exposées dans les paragraphes suivants.

Les rendus graphiques La possibilité d'avoir accès à des rendus graphiques divers et complets est primordiale pour la visualisation moléculaire. La complexité des molécules, le nombre important d'atomes, les nombreuses métainformations, les structures particulières nécessitent d'avoir à sa disposition des moyens évolués et variés pour afficher une molécule. Quatre représentations différentes (voir figure 2.2 page ci-contre) ont été utilisées sur la plate-forme SHADDOCK :

CPK affiche tous les atomes de la molécule sous forme de sphères en les reliant par des cylindres ; c'est un affichage très chargé lorsque le nombre d'atomes est importants mais on peut modifier la taille des sphères et des cylindres (voir figure 2.2a page suivante) ;

Licorice représente tous les liens entre les atomes par des cylindres, sans représenter les atomes ; la taille des cylindres peut être modifiée (voir figure 2.2b page ci-contre) ;

NewRibbon produit une courbe spline sur les atomes C_α représentant l'armature principale de la molécule ; la courbe est représentée sous forme de ruban (voir figure 2.2c page suivante) ;

HBonds affiche les potentielles liaisons hydrogène sous forme de traits en pointillés; les seuils d’affichage ainsi que les paramètres de la ligne en pointillés (couleur, largeur, *etc.*) sont modifiables (voir figure 2.2d).

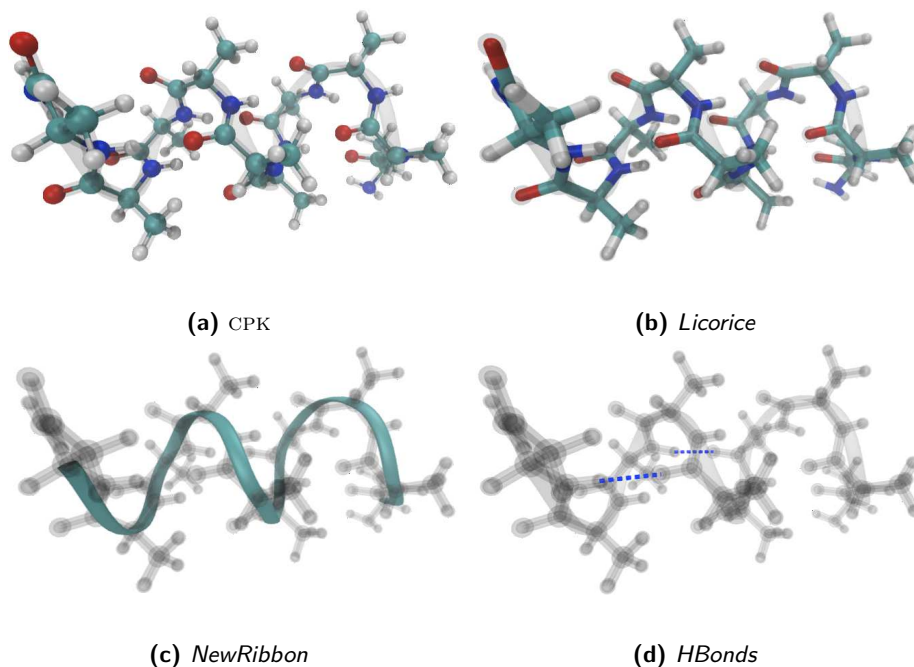


Figure 2.2 – Illustration des représentations de molécules sur VMD

Chacune de ces représentations visuelles peut être affectée à tout ou partie de la molécule comme par exemple « le résidu 13 », « seulement les atomes de carbone » ou « tous les résidus entre 1 et 16 sauf les atomes d’hydrogène ». De plus, pour chacune des représentations précédentes, différentes colorations sont possibles :

Couleur fixe donne une couleur unie prédéfinie (pour la couleur du curseur par exemple) ;

Couleur des atomes donne une couleur différente à chaque atome selon un code couleur standard dépendant de sa nature (rouge pour oxygène, blanc pour hydrogène, *etc.*) ;

Couleur des résidus donne une couleur différente pour chaque atome selon une palette de couleurs prédéfinie par VMD ;

Transparence rend transparent les objets tout en conservant la teinte ;

GoodSell accentuant les contours des objets sous le principe du *cell shading*.

Les outils de manipulation La manipulation des molécules est nécessaire sur la plate-forme SHADDOCK. VMD dispose déjà de différents outils permettant d'effectuer différentes manipulation sur les molécules.

Par défaut et sans configuration, la souris permet d'orienter la scène sur trois degrés de liberté (DDLs) afin d'observer la molécule sous différents angles.

Il est également possible d'utiliser une souris 3D, automatiquement détectée lorsqu'elle est branchée sur l'ordinateur. Une souris 3D permet de translater et d'orienter la scène. La souris 3D SpaceNavigator® est utilisée dans le cadre de certaines de nos expérimentations.

Enfin, des outils spécifiques sont disponibles par l'intermédiaire d'une connexion avec VRPN (*Virtual Reality Protocol Network*) (voir section 2.3.2 page 15). Ces outils sont liés à des périphériques externes (des interfaces PHANTOM Omni® dans notre cas). Les outils disponibles par défaut dans VMD ont été utilisés dans la première expérimentation (voir chapitre 3 page 23) et sont :

grab qui permet de sélectionner une molécule dans son intégralité et de la déplacer dans la scène ;

tug qui permet de sélectionner un atome de la molécule et de lui appliquer une force (qui sera transmise à la simulation).

Cependant, de nombreux outils supplémentaires ont été développés au-fur-et-à-mesure des besoins identifiés durant les expérimentations. Ces nouveaux outils sont détaillés dans la section 2.4 page 17.

La génération automatique de fichier de simulation La simulation nécessite de nombreuses informations dont l'ensemble des liaisons entre atomes, des angles simples, des angles diédraux et des angles de torsion. La simple description des atomes et de leurs positions (fichier PDB) couplée aux résultats de CHARMM [BROOKS et al. 1983] permet de générer les fichiers nécessaires à la simulation. VMD fournit tous les outils permettant de générer ce fichier nécessaire à la simulation (fichier PSF) par l'intermédiaire d'une extension : *Automatic PSF builder*.

NAMD (*Scalable Molecular Dynamics*)

Les deux logiciels de simulation principaux existants sont NAMD [PHILLIPS et al. 2005] et GROMACS [BERENDSEN et al. 1995]. Bien que GROMACS soit plus performant que NAMD, surtout dans les dernières versions [HESS et al. 2008] qui offre des performances jusqu'à quatre fois plus rapide que NAMD. Cependant, NAMD est développé par la même université que VMD et l'interaction entre les deux logiciels est donc extrêmement facilitée. De plus, les petites molécules que nous utiliserons lors de nos simulations ne nécessitent pas des performances exceptionnelles. Enfin, NAMD peut être aisément connecté à VMD dans le cadre d'une simulation interactive (voir section 2.2.2 page ci-contre) contrairement à GROMACS. C'est pourquoi le logiciel NAMD a été retenu pour notre plate-forme.

Une des fonctionnalités de `NAMD` utilisée est la possibilité de *fixer* des atomes. En effet, la fixation d'atomes permet d'exclure partiellement certains atomes durant la simulation. Ces atomes interviennent dans le calcul des forces de la simulation mais eux-mêmes ne sont pas soumis aux forces de l'environnement. Cette fonctionnalité est nécessaire pour simuler un point d'ancrage de la molécule dans l'environnement virtuel. Sans ce point d'ancrage, la molécule pourrait dériver et sortir de l'espace de travail des utilisateurs sans possibilité de récupération.

IMD (*Interactive Molecular Dynamics*)

Les logiciels de simulation ne sont pas développés pour des simulations interactives en temps-réel. Cependant, l'ITAP a développé le protocole `IMD` permettant d'utiliser `NAMD` couplé à `VMD` pour des simulations interactives en temps-réel [STADLER et al. 1997]. L'extension `IMD connect` permet de connecter rapidement le logiciel `VMD` avec la simulation de `NAMD`.

Cependant, entre le début du développement de notre plate-forme en 2008 et aujourd'hui, une nouvelle solution plus générique a été développée au sein de l'Institut de Biologie Physico-Chimie (IBPC). En effet, `MDDRIVER` [DELALANDE et al. 2009] est un petit logiciel permettant d'utiliser le protocole `IMD` avec l'autre logiciel de simulation `GROMACS`. Il se présente sous la forme d'une interface permettant de choisir le logiciel de simulation ainsi que le logiciel de visualisation. Il permet également de connecter plusieurs logiciels de visualisation à une même simulation. Cependant, cette nouvelle solution n'a pas encore été implémentée dans notre plate-forme mais c'est une amélioration technique qui sera effectuée dans les prochaines versions de la plate-forme.

Références

- BERENDSEN, Herman J. C., David van der SPOEL et Rudy van DRUNEN (1995). « `GROMACS` : a message-passing parallel molecular dynamics implementation ». Dans *Computer Physics Communications* 91.1–3, pages 43–56 (cf. page 12).
- BROOKS, Bernard R., Robert E. BRUCCOLERI, Barry D. OLAFSON, David J. STATES, Sundaramoorthi SWAMINATHAN et Martin KARPLUS (avr. 1983). « `CHARMM` : a program for macromolecular energy, minimization, and dynamics calculations ». Dans *Journal of computational chemistry* 4.2, pages 187–217 (cf. page 12).
- DELALANDE, Olivier, Nicolas FÉREY, Gilles GRASSEAU et Marc BAADEN (avr. 2009). « Complex molecular assemblies at hand via interactive simulations ». Dans *Journal of computational chemistry* 30.15, pages 2375–2387 (cf. page 13).

- HESS, Berk, Carsten KUTZNER, David van der SPOEL et Erik LINDAHL (nov. 2008). « GROMACS 4 : algorithms for highly efficient, load-balanced, and scalable molecular simulation ». Dans *Journal of chemical theory and computation* 4.3, pages 435–447 (cf. page 12).
- HUMPHREY, William F., Andrew DALKE et Klaus SCHULTEN (fév. 1996). « VMD : Visual Molecular Dynamics ». Dans *Journal of Molecular Graphics* 14.1, pages 33–38 (cf. page 10).
- PETTERSEN, Eric F., Thomas D. GODDARD, Conrad C. HUANG, Gregory S. COUCH, Daniel M. GREENBLATT, Elaine C. MENG et Thomas E. FER-
RIN (oct. 2004). « UCSF Chimera - a visualization system for exploratory research and analysis ». Dans *Journal of computational chemistry* 25.13, pages 1605–1612 (cf. page 10).
- PHILLIPS, James C., Rosemary BRAUN, Wei WANG, James GUMBART, Emad TAJKHORSHID, Elizabeth VILLA, Christophe CHIPOT, Robert D. SKEEL, Laxmikant KALÉ et Klaus SCHULTEN (mai 2005). « Scalable molecular dynamics with NAMD ». Dans *Journal of computational chemistry* 26.16, pages 1781–1802 (cf. page 12).
- SAYLE, Roger A. et E. James MILNER-WHITE (sept. 1995). « RASMOL : biomolecular graphics for all ». Dans *Trends in biochemical sciences* 20.9, pages 374–376 (cf. page 10).
- SCHRÖDINGER, LLC, éditeur (août 2010). *The PyMOL molecular graphics system*. URL : <http://www.pymol.org/> (visité le 03/06/2011) (cf. page 10).
- SOURCEFORGE.NET, éditeur (août 2006). *Jmol : an open-source Java viewer for chemical structures in 3D*. URL : <http://www.jmol.org/> (visité le 03/06/2011) (cf. page 10).
- STADLER, Jörg, Ralf MIKULLA et Hans-Rainer TREBIN (1997). « IMD : a software package for molecular dynamics studies on parallel computers ». Dans *International Journal of Modern Physics* 8.5, pages 1131–1140 (cf. page 13).

2.3 Interaction avec la simulation moléculaire

La plate-forme SHADDOCK implémente également un moyen d’interagir avec cette simulation en temps-réel. Les besoins particuliers pour la manipulation sont présentés dans la section 2.3.1 puis les solutions logicielles dans la section 2.3.2 page ci-contre.

2.3.1 Les besoins

Afin de pouvoir modifier la simulation en temps-réel, il faut pouvoir manipuler des éléments de la molécule. Il faut donc être capable de sélectionner les atomes puis de leur appliquer une force. L’environnement virtuel est en 3D et

il paraît donc nécessaire d'avoir un périphérique de manipulation en 3D. Cette manipulation sera effectuée par le PHANTOM Omni® (voir section 2.3.2).

De plus, SHADDOCK est un EVC : il doit pouvoir accueillir plusieurs utilisateurs. La connexion de plusieurs périphériques apporte des contraintes matérielles et logicielles. Une architecture sous forme client/serveur sera assurée grâce au logiciel VRPN (voir section 2.3.2).

2.3.2 Interaction par interface haptique

PHANTOM Omni® et OPENHAPTICS™

Une plate-forme de simulation interactive en temps-réel nécessite des outils d'interaction. Parmi les périphériques d'interaction existants, il faut choisir un périphérique permettant au minimum six DDLs en entrée et au minimum trois DDLs en retour haptique. En effet, l'outil *grab* nécessite six DDLs en entrée et l'outil *tug* nécessite trois DDLs en entrée et trois DDLs en retour haptique. L'interface PHANTOM Omni® [MASSIE et SALISBURY 1994] de l'entreprise SENSABLE répond aux attentes de la plate-forme SHADDOCK (voir figure 2.3).



Figure 2.3 – Interface PHANTOM Omni® 6 DDL/3 DDL

À l'origine, les interfaces haptiques de SENSABLE étaient programmables à l'aide de l'interface de programmation (API) GHOST [SENSABLE 2002]. Le travail de ITKOWITZ et al. [2005] a permis de fournir une nouvelle API plus facile à utiliser : OPENHAPTICS™. C'est à partir de cette API que les interfaces haptiques sont utilisées sur SHADDOCK.

VRPN (*Virtual Reality Protocol Network*)

VMD offre un moyen simple et relativement universel de connecter un périphérique. En effet, il gère les connexions de périphériques par l'intermédiaire

de VRPN [TAYLOR II et al. 2001]. VRPN fonctionne sous la forme d’une architecture client/serveur où VMD est l’application cliente. L’interface haptique est connectée physiquement à un autre ordinateur (le même ordinateur le cas échéant) et un serveur VRPN communique avec cette interface. C’est seulement par l’intermédiaire de VRPN et à travers le réseau *Ethernet* que VMD va percevoir les mouvements de l’interface haptique et lui envoyer les efforts à fournir. La compilation de VRPN en tant que serveur de PHANTOM Omni® sous le système d’exploitation LINUX (UBUNTU v10.04) a demandé quelques modifications dans le code. Ces modifications ont été soumises au développeur de VRPN qui les a intégrées dans les dernières versions.

L’avantage de cette architecture est la possibilité d’ajouter autant de serveurs et donc autant d’interfaces haptiques que voulu. Cependant, cela suppose également d’avoir autant d’ordinateurs que de serveurs ce qui complique une logistique complexe. On pourra noter que la chaleur dégagée par l’ensemble de ces machines additionnée à celle du vidéo-projecteur provoque des conditions d’expérimentation rapidement désagréables. C’est pourquoi aucune des expérimentations proposées ne durait plus de 30 mn ou, le cas échéant, une pause est effectuée au bout de 30 mn afin d’aérer la salle d’expérimentation.

Références

- ITKOWITZ, Brandon, Josh HANDLEY et Weihang ZHU (mar. 2005). « The OpenHaptics™ toolkit : a library for adding 3D™ navigation and haptics to graphics applications ». Dans *Proceedings of the first joint eurohaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems*. WHC '05. Washington, DC, USA : IEEE Computer Society, pages 590–591 (cf. page 15).
- MASSIE, Thomas H. et Kenneth J. SALISBURY (1994). « PHANTOM haptic interface : a device for probing virtual objects ». Dans *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Tome 55. 1. Massachusetts Institute of Technology, Cambridge, United States, pages 295–300 (cf. page 15).
- SENSABLE, Technologies Incorporated® (avr. 2002). *GHOST® software developer’s toolkit programmer’s guide*. 4.0. SensAble Technologies Incorporated® (cf. page 15).
- TAYLOR II, Russell M., Thomas C. HUDSON, Adam SEEGER, Hans WEBER, Jeffrey JULIANO et Aron T. HELSER (2001). « VRPN : a device-independent, network-transparent VR peripheral system ». Dans *Proceedings of the ACM symposium on Virtual reality software and technology*. Virtual Reality Software and Technology 2001. New York, NY, USA : ACM, pages 55–61 (cf. page 16).

2.4 Nouveaux outils pour l'interaction

Durant les différentes études présentées dans la [partie II page 23](#), les analyses et les remarques d'utilisateurs ont permis d'améliorer les outils d'interaction et d'en proposer de nouveaux. Le développement de ces nouveaux outils a nécessité une modification du programme `VMD` par extension des outils déjà existants. Des fonctionnalités ont été ajoutées et sont présentées dans les sections suivantes.

2.4.1 Amélioration de la sélection

Durant le processus de recherche et de sélection, les utilisateurs ont souvent évoqué le besoin de connaître en continu leur position et de savoir à priori l'élément qui va être sélectionné. Pour que les utilisateurs connaissent à chaque instant l'élément qui peut être sélectionné, une information visuelle met en surbrillance l'élément pointé à chaque instant. La mise en surbrillance est un agrandissement en transparence de l'élément pointé. La couleur de cette mise en surbrillance est de la même couleur que le curseur de l'utilisateur.

Cependant, lorsque seul un atome est mis en surbrillance, il peut être difficile de l'apercevoir. En effet, le nombre important d'atomes d'une molécule peut surcharger le rendu graphique. C'est pourquoi, l'ensemble du résidu auquel appartient l'atome pointé est également mis en surbrillance. Seul l'atome pointé est agrandi.

Une fois l'élément pointé, l'utilisateur peut sélectionner l'élément. Lorsque les utilisateurs sélectionnent l'élément, la surbrillance passera de la transparence à l'opacité. Une illustration des effets visuels relatifs au pointage et à la sélection est affichée sur la [figure 2.4](#).

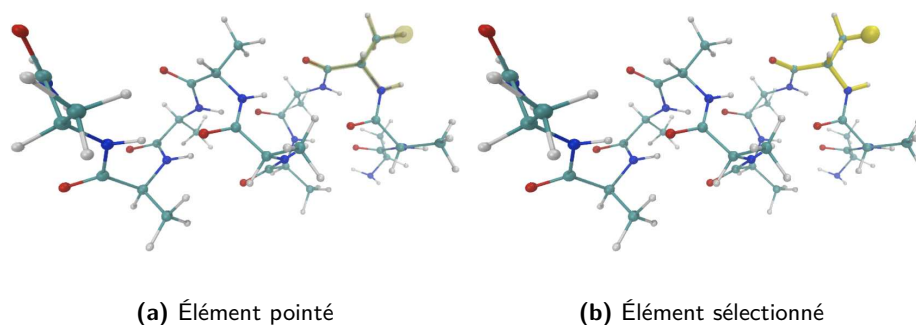


Figure 2.4 – Différence visuelle entre les éléments pointés et sélectionnés

2.4.2 Déformation par groupe d'atomes

L'outil *tug* permet de déformer la molécule en appliquant un effort à l'atome sélectionné. Cependant, la déformation par l'intermédiaire d'un seul atome pos-

sède deux désavantages.

Tout d’abord, la déformation d’une molécule atome par atome est un processus très fastidieux. Il serait plus efficace de déplacer un groupe d’atomes en une seule fois.

De plus, l’application d’un effort sur un atome provoque l’étirement de la molécule. Au repos, la molécule est dans état relativement stable. Déplacer un atome perturbe cet état de stabilité. De plus, certains atomes sont fortement liés et les éloigner peut perturber grandement l’état de stabilité. Il est donc préférable de déplacer tous ces atomes liés en une seule manipulation.

C’est pourquoi un outil appliquant un effort à un groupe d’atomes permet de déplacer un bloc d’atomes tout en conservant une certaine stabilité. Les groupes d’atomes dignes d’intérêt sont les résidus (une vingtaine d’atomes), les hélices- α ou feuillets- β (une vingtaine de résidus) et les molécules. Cependant, `VMD` n’est pas capable de fournir l’information sémantique regroupant les atomes en hélices- α ou en feuillets- β . La fonctionnalité de l’outil *tug* a donc été étendue aux résidus et aux molécules.

Cependant, appliquer le même effort à l’ensemble des atomes de la molécule produit un effort total très important. Si l’effort total est trop important, les perturbations envoyées à la simulation sont trop importantes et produisent des incohérences dans la simulation voire même un arrêt de la simulation. Il est donc nécessaire de diviser l’intensité des forces proportionnellement au nombre d’atomes sélectionnés.

2.4.3 Outil de désignation et attraction

Un aspect récurrent constaté durant les expérimentations est la nécessité de désigner un élément de la molécule. Parfois les utilisateurs éprouvent le besoin de désigner mais la plupart du temps, ce sont les enregistrement audio qui ont permis d’identifier ce besoin.

L’outil de désignation a été conçu pour répondre à un processus en quatre étapes :

- a. Recherche d’une cible (voir figure 2.5a page ci-contre) ;
- b. Désignation d’une cible (voir figure 2.5b page suivante) ;
- c. Acceptation d’une cible (voir figure 2.5c page ci-contre) ;
- d. Sélection d’une cible (voir figure 2.5d page suivante).

L’ étape a consiste pour un utilisateur \mathcal{A} à rechercher une cible à désigner. La recherche de cette cible répondra au besoin d’obtenir de l’aide d’un utilisateur \mathcal{B} .

L’ étape b consiste pour l’utilisateur \mathcal{A} à désigner la cible identifiée. La cible est alors mise en surbrillance de façon à être vue des autres utilisateurs.

L’ étape c page ci-contre fait intervenir l’utilisateur \mathcal{B} . L’utilisateur \mathcal{B} peut accepter ou non cette désignation. S’il accepte la désignation, la cible est alors

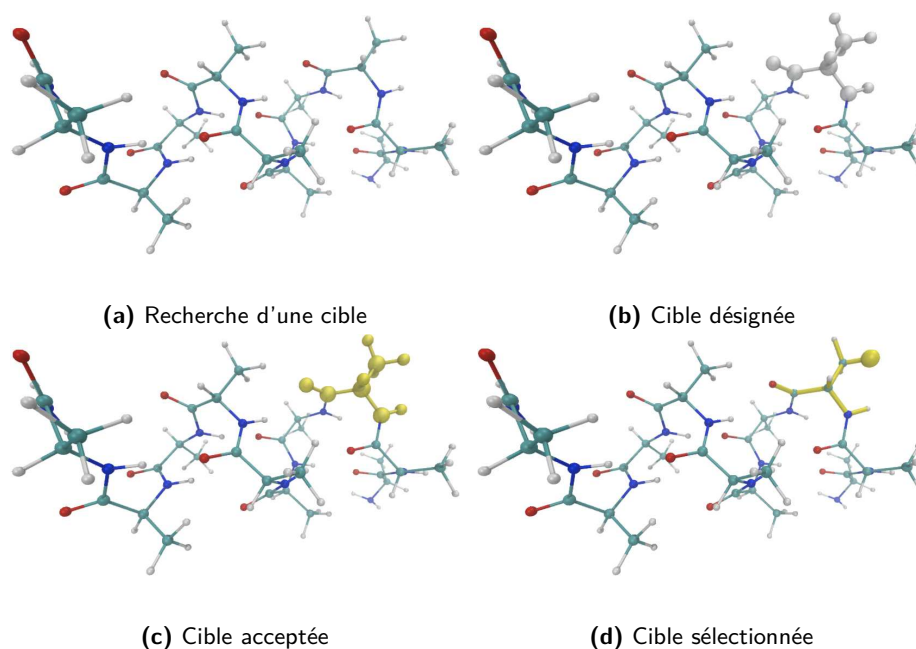


Figure 2.5 – Les quatre étapes de la désignation

colorée de la couleur du curseur de l'utilisateur \mathcal{B} qui a accepté. Tant qu'elle n'est pas acceptée, la résidu reste en surbrillance jusqu'à ce que la requête soit acceptée ou modifiée.

L' étape d page précédente est la dernière étape. L'utilisateur \mathcal{B} ayant accepté doit maintenant sélectionner la cible pour achever le processus de désignation. Tant que l'utilisateur \mathcal{B} n'a pas sélectionné le résidu ciblé, le processus ne peut pas être considéré comme terminé et l'effet de surbrillance reste actif.

Parallèlement à ces quatre étapes, des aides haptiques ont été ajoutées pour la dernière expérimentation (voir chapitre 6 page 61). Pour l' étape b page précédente, des vibrations sont générées sur tous les utilisateurs concernés par la désignation. De plus, dès l'instant qu'un utilisateur a accepté la désignation (voir étape c page ci-contre), il est guidé de façon haptique vers la cible. La vibration chez tous les autres utilisateurs est arrêtée.

Glossaire

amarrage moléculaire

Méthode permettant de déterminer l'orientation et la déformation optimale de 2 molécules afin qu'elle s'assemble pour former un complexe stable. iii, 3, 61, 64, 71

bimanuel

Qui se fait avec les deux mains. 47, 48, 54

binôme

Groupe constitué de 2 personnes. iv, vii, 23–27, 30, 32, 33, 35–37, 39–41, 44, 47–51, 54, 64

curseur

Élément virtuel associé à un élément physique que le sujet manipule ; il est lié à l'effecteur terminal. 71

docking moléculaire

Voir amarrage moléculaire. 3

effecteur terminal

Élément physique que le sujet manipule ; il est lié au curseur du monde virtuel. 26, 50, 63

homoscedasticité

Équivalent à homogénéité des variances ; permet de comparer des variables aléatoires possédant des variances similaires. 33

monôme

Groupe constitué d'une unique personne. vii, 23–26, 30, 32, 33, 35, 44, 47–51

résidu

Groupe d'atomes constituant un des blocs élémentaires d'une molécule. vii, ix, 4, 10, 17, 18, 26, 28, 30, 32–36, 39–41, 44, 45, 50–54, 63, 65

variable dépendante

Facteur mesuré sur une expérimentation (nombre de sélections, trajectoire, etc.) ; ces variables sont influencées par les variables indépendantes. 62, 71

variable indépendante

Facteur pouvant varier et être manipuler sur une expérimentation (nombre

de participants, tâche, *etc.*); ces variables vont avoir une incidence sur les variables dépendantes. 26, 50, 63, 71

variable inter-population

Variables pour lesquelles les sujets sont confrontés à une et une seule des modalités de la variable. 50

variable intra-population

Variables pour lesquelles les sujets sont confrontés à toutes les modalités de la variable. 26, 33, 50, 63

Acronymes

API

API vient de l'anglais *Application Programming Interface* et désigne une interface avec un programme informatique. 15

DDL

Mouvements relatifs indépendants d'un solide par rapport à un autre. vii, 11, 14, 15

EVC

Ensemble logiciel et matériel permettant de faire interagir plusieurs utilisateurs au sein d'un même environnement ; ils jouent un rôle important dans le développement de nouvelles méthodes de travail collaboratives. 5–7, 14, 24, 28, 45, 48, 51, 62, 64

IBPC

Institut de recherche, géré par la fédération de recherche FRC 550, étudiant les bases structurales, génétiques et physico-chimiques à leur différents niveaux d'intégration. 13, 62

IMD

Programme permettant de connecter le logiciel de visualisation moléculaire `VMD` avec le logiciel de simulation `NAMD` pour une simulation interactive en temps-réel [STADLER et al. 1997]. 5, 9, 12, 13, 73

ITAP

Institut de Physique Théorique et Appliquée de STUTTGART à l'origine du développement du logiciel `IMD`. 9, 12

CNRS–LIMS

Unité Propre de Recherche du CNRS (UPR 3251) associé aux universités PARIS Sud et Pierre et Marie CURIE. i, 25, 49

NAMD

Programme de simulation pour la dynamique moléculaire [PHILLIPS et al. 2005]. 5, 9, 12, 64, 73

RMSD

Appelé Écart Quadratique Moyen en français, il permet – dans le cadre de la biologie moléculaire – de mesurer la différence entre deux déformations d'une même molécule. 52, 64

Acronymes

VMD

Programme de visualisation moléculaire [HUMPHREY et al. 1996]. vii, 5, 9–12, 15–17, 73

VRPN

Logiciel permettant de connecter différents périphériques de réalité virtuelle à une même application sous forme d’une architecture client/serveur [TAYLOR II et al. 2001]. 5, 11, 14, 15