

Projet M1 ISICG
Ray tracing de molécules

Encadrants :

- Maxime MARIA (maxime.maria@unilim.fr)
Maître de Conférences en Informatique à l’Université de Limoges - Institut de recherche **XLIM**, équipe **ASALI/SIR**
Chercheur associé au laboratoire **GBCM**, équipe **M2D2** - CNAM Paris
- Simon GUIONNIERE
Ingénieur d’études, laboratoire **GBCM**, équipe **M2D2** - CNAM Paris

1 Introduction

Le domaine de la conception de médicaments implique fréquemment l’utilisation de petites molécules chimiques, sélectionnées sur la base de leur potentiel modulateur de l’activité de la cible pharmacologique. Cette sélection peut être effectuée via différentes méthodes de simulation, telles que les méthodes de *docking* qui permettent de prédire la géométrie du complexe entre la molécule candidate et la protéine cible. Une visualisation précise des molécules, de leurs propriétés et de leurs interactions physico-chimiques est nécessaire pour comprendre le mécanisme d’action des molécules candidates et ainsi proposer des modifications chimiques pour en synthétiser des meilleures. Ces étapes de simulation font désormais partie intégrante des processus de développement de nouvelles molécules à visée thérapeutique.

La visualisation scientifique de molécules repose sur différentes représentations géométriques, dont l’utilité dépend du type de simulation. La figure 1 en donne quelques exemples. Pour plus de détails concernant ces représentations, nous nous référerons à [KKF⁺17].

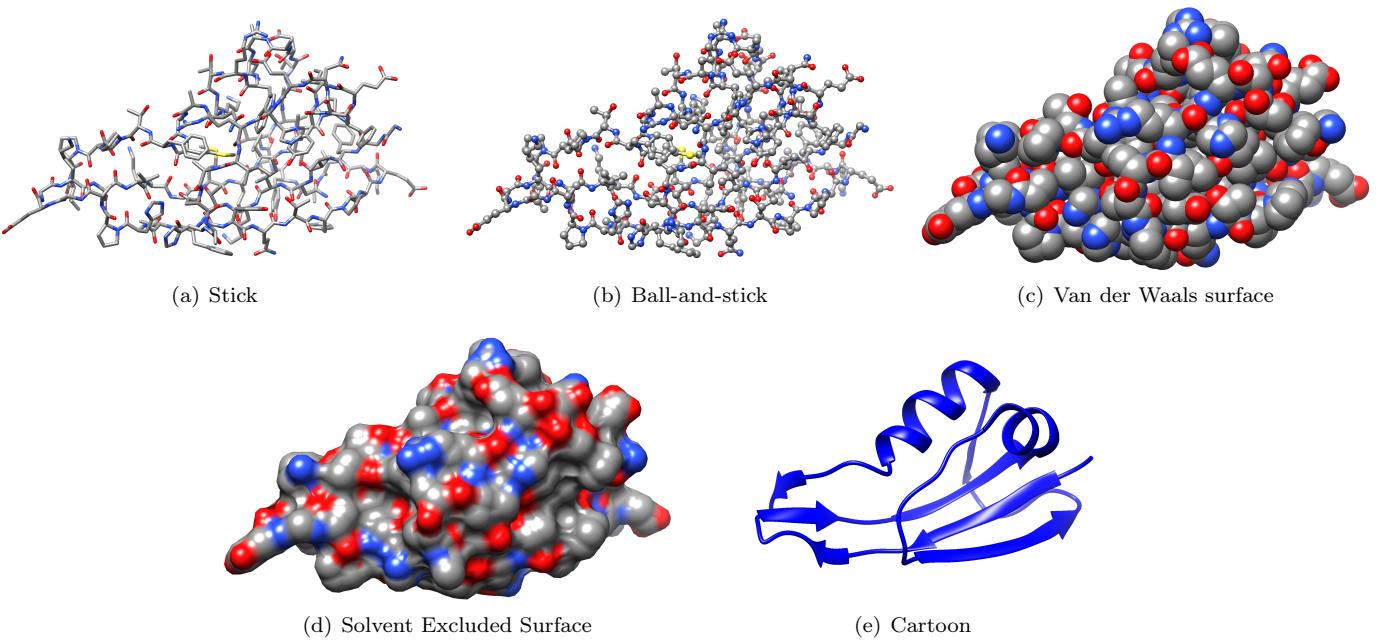


FIGURE 1 – Plusieurs représentations d’une même petite protéine (3IGE).

2 Objectifs

Dans ce projet, vous devrez développer un moteur de ray-tracing « offline » (*i.e.* qui calcule une seule image de la scène) spécifiquement dédié à la visualisation de systèmes moléculaires. En fonction des performances que vous atteindrez, nous pourrons éventuellement imaginer un rendu temps-réel, mais ce n’est pas l’objectif principal de ce projet.

Vous vous concentrerez uniquement sur les représentations composées de sphères et/ou de cylindres (*cf.* figures 1(a), 1(b) ou 1(c)). L’intersection entre un rayon et ce type de primitives géométriques est rapide à calculer et simple à mettre en œuvre. C’est pourquoi, nous nous passerons de l’utilisation de maillages afin d’avoir de meilleures performances de rendu, de consommer moins de mémoire et de permettre un rendu parfait au pixel près. En effet, quelque soit la qualité du maillage celui-ci est visible au moins à un certain niveau de zoom (*cf.* figure 2).

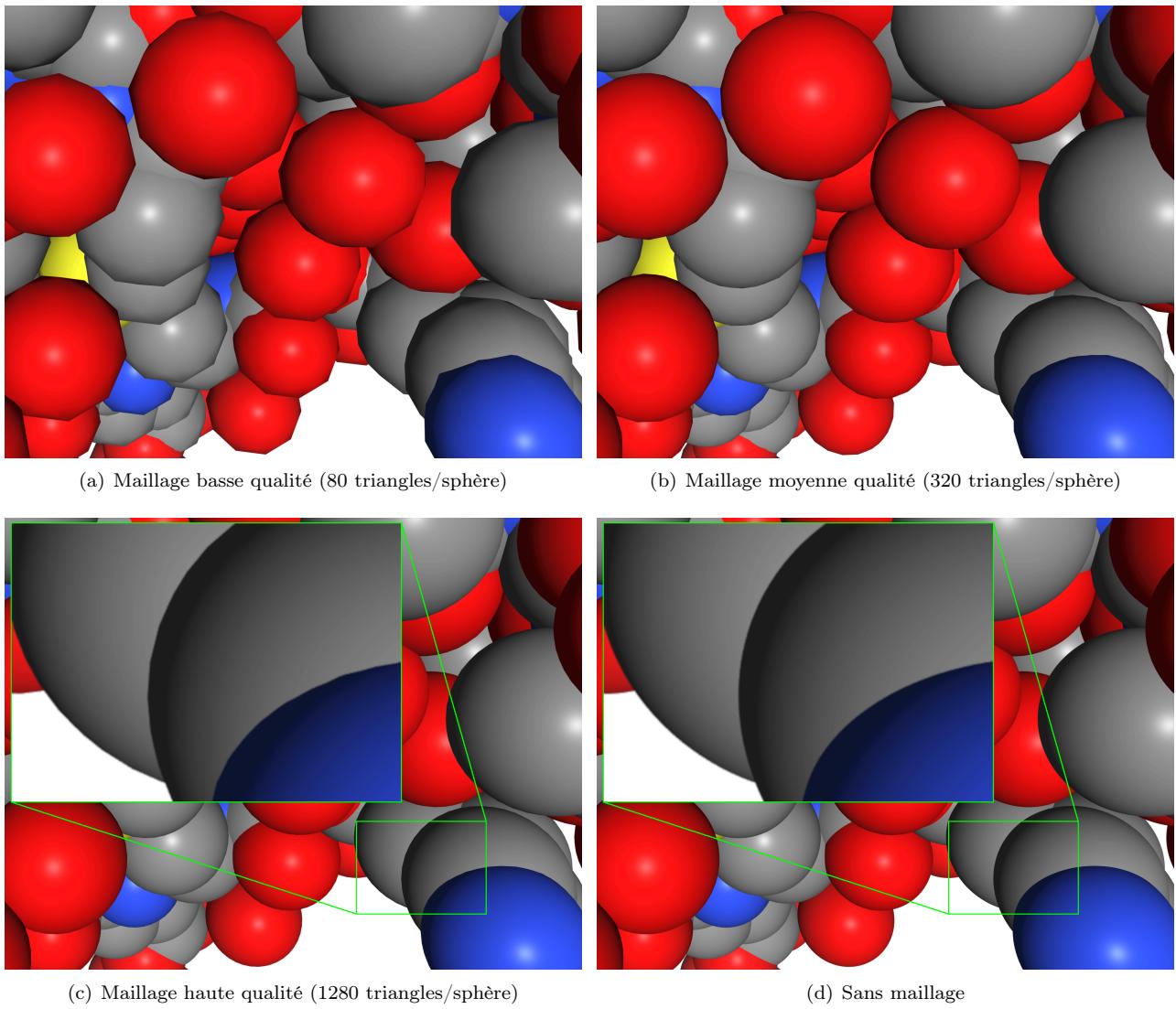


FIGURE 2 – Comparaison de la qualité de rendu de sphères pour différents niveaux de subdivision du maillage et sans utiliser de maillage.

2.1 Fonctionnalités attendues

Votre moteur de rendu devra être codé en C++ et présenter **au moins** les fonctionnalités suivantes :

1. Calcul d'une image de molécules à partir d'une scène et d'une position de caméra données
2. Possibilité de choisir entre la représentation de Van der Walls (ensembles de sphères, figure 1(c)) et la Ball-and-stick (ensemble de sphères et de cylindres, figure 1(b))
3. Calcul de l'éclairage direct par une ou plusieurs sources lumineuses ponctuelles
4. Possibilité de choisir entre un matériau diffus et spéculaire
5. Utilisation d'une structure accélératrice de type « grille régulière » pour réduire les temps de calcul

La difficulté majeure du projet sera d'implémenter la structure accélératrice connue comme le « Fitted-BVH » qui permet de faire un rendu de « meta-balls » à partir des atomes [GPP⁺10]. Un exemple de rendu est proposé sur la figure 3.

Nous vous fournirons le code permettant de charger une structure moléculaire.

2.2 Extensions possibles

Une fois que toutes les fonctionnalités attendues seront implémentées, testées et validées, vous pourrez améliorer votre programme en proposant par exemple :

- Ajout d'une *skybox*

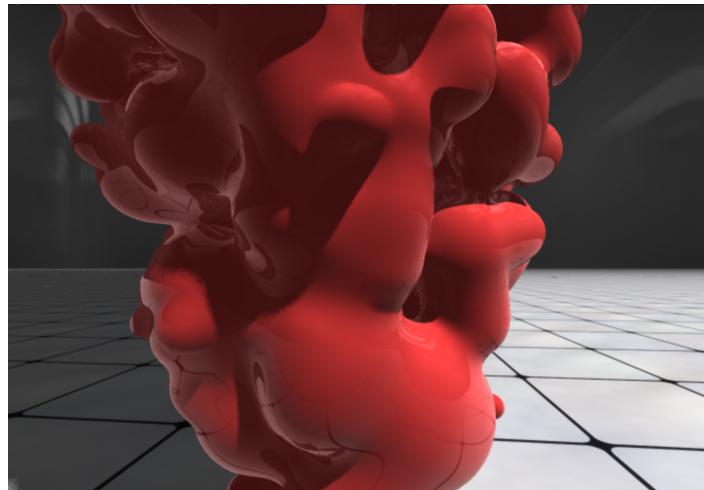


FIGURE 3 – Rendu de meta-balls

- Utilisation de textures à plaquer sur les primitives géométriques
- D'autres matériaux *e.g.* du « verre » (il faudra alors simuler les phénomènes de réflexion et de réfraction)
- D'autres représentations *e.g.* les hyperballs
- Calcul des inter-réflexions lumineuses
- Calcul de l'éclairage global de la scène

Attention certaines de ces propositions sont beaucoup plus complexes à mettre en œuvre que d'autres. Avant de faire un choix, contactez-nous ! Nous pourrons vous expliquer certains principes fondamentaux, vous aiguiller dans vos recherches et évaluer la faisabilité de l'extension.

Si vous avez d'autres idées, surtout, n'hésitez pas ! Par contre, encore une fois, contactez-nous d'abord... Ce serait dommage de travailler sur quelque chose que nous trouvons sans intérêt ou qui serait infaisable dans le temps imparti.

Références

- [GPP⁺10] Olivier Gourmel, Anthony Pajot, Mathias Paulin, Loïc Barthe, and Pierre Poulin. Fitted BVH for Fast Raytracing of Metaballs. *Computer Graphics Forum, Eurographics 2010 Proceedings*, 29(2) :281–288, mai 2010.
- [KKF⁺17] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of biomolecular structures : State of the art revisited. *Computer Graphics Forum*, 36(8) :178–204, 2017.