



HAL
open science

Construction efficace de la surface de systèmes moléculaires complexes sur GPU

Cyprien Plateau–Holleville, Matthieu Montes, Maxime Maria, Stéphane
Mérillou

► **To cite this version:**

Cyprien Plateau–Holleville, Matthieu Montes, Maxime Maria, Stéphane Mérillou. Construction efficace de la surface de systèmes moléculaires complexes sur GPU. J.FIG 2023 - Journées Françaises de l'Informatique Graphique, Nov 2023, Montpellier, France. hal-04633888

HAL Id: hal-04633888

<https://hal.science/hal-04633888v1>

Submitted on 3 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Construction efficace de la surface de systèmes moléculaires complexes sur GPU

C. Plateau-Holleville¹, M. Montes^{2,3}, M. Maria¹, S. Mérellou¹

¹Laboratoire XLIM, UMR CNRS 7252, Université de Limoges, France

²GBCM - Laboratoire Génomique, Bioinformatique et Chimie Moléculaire, EA7528, Conservatoire National des Arts et Métiers, Hésam Université, France

³Institut Universitaire de France (IUF)

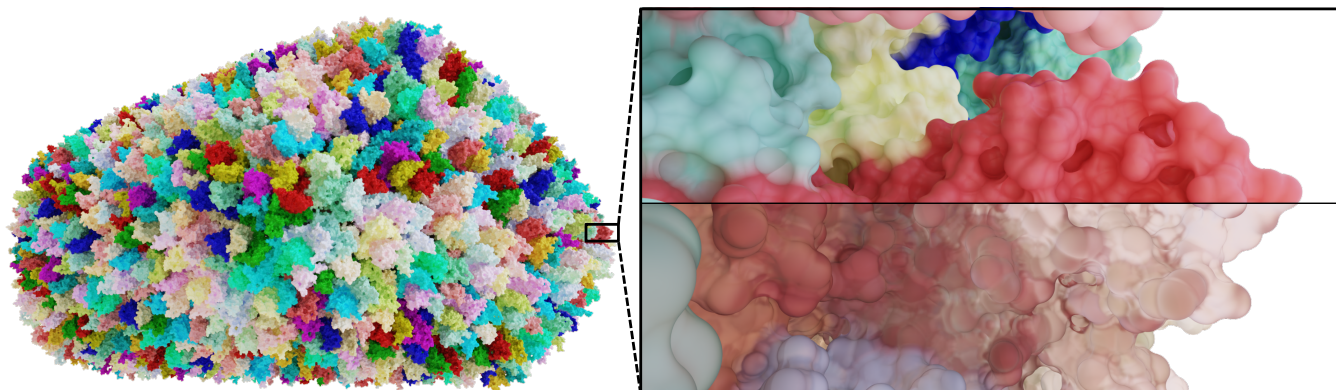


Figure 1: Rendu de la Surface Exclue au Solvant (SES) de la capsid complète du VIH-1 (PDB : 3J3Q) de plus de 2.4 millions d'atomes. Sa géométrie est calculée avec notre méthode en 700ms sur une NVIDIA RTX 2080. Comme illustré avec cette grande structure, la SES permet de représenter de façon précise les détails locaux tout en conservant une bonne lisibilité des formes générales à plus grande échelle.

Abstract

La surface exclue au solvant est une représentation moléculaire intéressante dans un cadre de visualisation grâce à sa capacité à illustrer les potentielles interactions. Son calcul reste cependant difficile en raison de la complexité de sa définition géométrique, mais aussi de la difficulté à paralléliser les opérations requises. Dans cet article, nous présentons une méthode de génération de cette surface, basée sur une récente analyse mathématique, permettant de s'affranchir des limitations précédentes, notamment grâce à des traitements GPU adaptés.

CCS Concepts

•**Computing methodologies** → Massively parallel algorithms; Parametric curve and surface models;

1. Introduction

La visualisation moléculaire est un outil particulièrement intéressant pour l'analyse et la compréhension de la géométrie des protéines. À travers le développement d'outils adaptés et depuis son introduction, l'informatique graphique a contribué à de multiples reprises aux outils disponibles pour faciliter la visualisation, mais aussi à l'illustration et la mise en valeur des observations [KKF*16].

Avec l'amélioration des méthodes d'acquisition, les études de plus grandes échelles sont devenues plus courantes, mettant à l'épreuve les stratégies de visualisation existantes. Même si de ré-

centes méthodes ont permis d'utiliser la répétition de motifs au sein de la géométrie afin d'améliorer les performances de visualisation de très grandes structures [MAPV15], le support de dynamiques moléculaires empêche tout à priori géométrique.

Dans ce contexte et depuis sa définition analytique par Richards [Ric77], la Surface Exclue au Solvant (SES) s'est installée comme une représentation essentielle dans le travail d'analyse des biochimistes. Elle est définie par l'interaction entre la surface de van der Waals (vdW), où chaque atome est représenté par une sphère d'un rayon égal à son rayon de vdW, et une sphère appelée

21 probe, comme illustrée sur la figure 2. Elle décrit donc l'interaction
 22 potentielle avec d'autres entités comme des solvants ou autres
 23 molécules. Ainsi, elle est utile dans des contextes de tailles de pro-
 24 téines très variées, comme illustrée sur la figure 1, pour la visuali-
 25 sation de caractéristiques externes, mais aussi internes comme des
 26 cavités au sein de la géométrie.

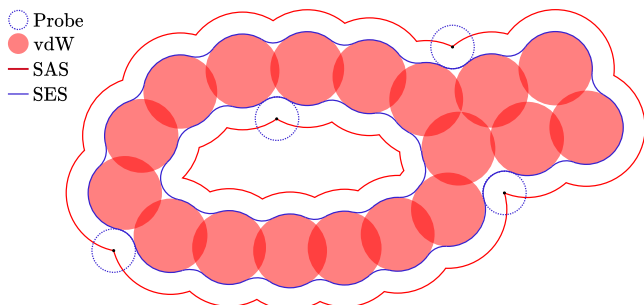


Figure 2: Illustration de la surface de van der Waals (vdW), de la Surface Accessible au Solvant (SAS) et de la Surface Exclue au Solvant (SES). Le centre du probe, en roulant sur la surface de vdW, décrit la SAS tandis que sa frontière avec la surface de vdW décrit la SES.

27 Le calcul analytique de la SES pour de grandes structures en
 28 des temps réduits pour le support de dynamiques moléculaires reste
 29 difficile malgré différents travaux originaux. Dans cet article, nous
 30 présentons une méthode adaptée à ces contraintes à partir des contri-
 31 butions suivantes :

- 32 • Nous proposons l'utilisation de la dernière analyse mathéma-
 33 tique de la SES [QS16, QS17] permettant de diminuer les besoins
 34 mémoires requis pour les calculs.
- 35 • Nous proposons un processus de construction adapté aux archi-
 36 tectures modernes de GPU permettant le calcul de la surface
 37 complète dans des temps similaires à la méthode la plus rapide
 38 de l'état de l'art, elle-même limitée à la surface extérieure.
- 39 • Nous montrons qu'à partir de nos développements, il est possi-
 40 ble de calculer la SES de grandes structures sur des cartes
 41 graphiques grand public.

42 2. Travaux antérieurs

43 À partir des travaux introductifs de Richards [Ric77], différentes
 44 études ont été présentées sur la SES. Nous proposons de les diviser
 45 en trois catégories : fondamentale, approximant et analytique. Le
 46 premier type s'est principalement concentré sur l'analyse de la sur-
 47 face et la construction d'algorithmes théoriques, tandis que les deux-
 48 ième et troisième se sont intéressés à sa construction de façon adap-
 49 tée à des besoins de visualisation.

50 **Analyses fondamentales** Les premiers éléments fondamentaux
 51 de calculs de la SES ont été présentés par Connolly [Con83]. C'est
 52 sur cette base que deux algorithmes, Reduced-Surface [SOS96] et
 53 Contour-Buildup [TA96], ont été introduits. Le premier est basé
 54 sur un traitement des cercles résultant de l'intersection des sphères
 55 des atomes alors que le second propose de construire une α -
 56 shape [EM94] à partir de laquelle la surface peut être reconstruite.
 57 Enfin, Quan & Stamm [QS16, QS17] ont plus récemment proposé

58 une analyse mathématique de la surface permettant de s'affranchir
 59 des singularités géométriques des précédentes représentations.

60 **Approximant** Les besoins en visualisation rapide de la SES
 61 ont conduit au développement de méthodes l'approximant. Notam-
 62 ment, l'utilisation de champs scalaire a permis l'introduction de vi-
 63 sualisation interactive [HKG*17]. D'autres surfaces moléculaires
 64 ont été aussi considérées, comme la surface gaussienne [Bru19],
 65 approchant la forme générale de la SES et permettant de meilleure
 66 performance, mais sans bénéficier d'un niveau de détail similaire.

67 **Analytique** Les liens induits entre consommation mémoire
 68 et niveau de détail des méthodes approximant ont poussé les
 69 développements des stratégies visant le calcul analytique de la
 70 surface. Notamment, des méthodes performantes de visualisation
 71 de l'extérieur de la SES [KBE09] ont été proposées en masquant
 72 certaines parties non souhaitées à l'intérieur du volume de la
 73 surface. Des implémentations parallèles d'algorithmes fondamen-
 74 taux [KDE10, KGE11] ont été développées sur cette base pouvant
 75 cependant nécessiter de grandes allocations mémoires. Des travaux
 76 permettant le calcul de la géométrie complète de la surface ont en-
 77 suite été présentés afin de permettre la représentation de l'intérieur
 78 de la surface [JPSK16, RZK*19], mais proposant des temps de cal-
 79 cul et une consommation mémoire plus importants.

80 3. Représentation géométrique

81 Notre méthode repose sur la représentation géométrique de la SES
 82 proposée par Quan & Stamm [QS16]. Celle-ci est basée sur un dé-
 83 coupage de la surface en primitives, appelée patches, reposant sur
 84 des éléments géométriques de la SAS à partir desquels on peut
 85 définir les fonctions distance signée (SDF) illustrées sur la figure 3.
 86 Cette description permet de définir une structure de données com-
 87 pacte permettant le rendu des patches que nous décrivons dans cette
 88 partie. Les SDFs sont données dans l'annexe A.

89 **Primitives de la SAS.** Comme illustré dans les figures 2 et 3, on
 90 peut définir les patches de la SES à partir de primitives géométriques
 91 de la SAS. Cette dernière est décrite comme l'union des sphères de
 92 vdW dont le rayon aurait été agrandi par celui du probe. Le chemin
 93 du probe est donc décrit par la surface de la SAS que l'on peut
 94 découper en trois types de primitives : la surface des sphères de la
 95 SAS, les cercles dérivés de l'intersection de deux sphères de la SAS
 96 et enfin les intersections de trois sphères de la SAS en un ou deux
 97 points.

98 **Patch convexe P_+ .** Un patch convexe propose deux degrés de
 99 liberté au probe roulant sur sa surface. Il peut être représenté avec
 100 une sphère de vdW dont les limites sont décrites par les cercles
 101 de la SAS. Un patch P_+ est totalement représenté à partir d'une
 102 position, un rayon et une liste de cercles de la SAS.

103 **Patches toroïdaux $P_{t,f}$ et P_t .** Lorsque le centre du probe est local-
 104 isé sur un cercle de la SAS, il dispose d'un seul degré de liberté. La
 105 surface décrite par la frontière entre le probe et la surface de vdW
 106 est alors similaire à celle de l'intérieur d'un tore. Puisqu'un cercle
 107 de la SAS peut être intersecté par un autre, résultant en une inter-
 108 section de la SAS, nous différencions les cercles complets donnant
 109 des patches toroïdaux complets $P_{t,f}$ et les cercles intersectés donnant
 110 des patches toroïdaux partiels P_t . Ils sont représentables à partir des

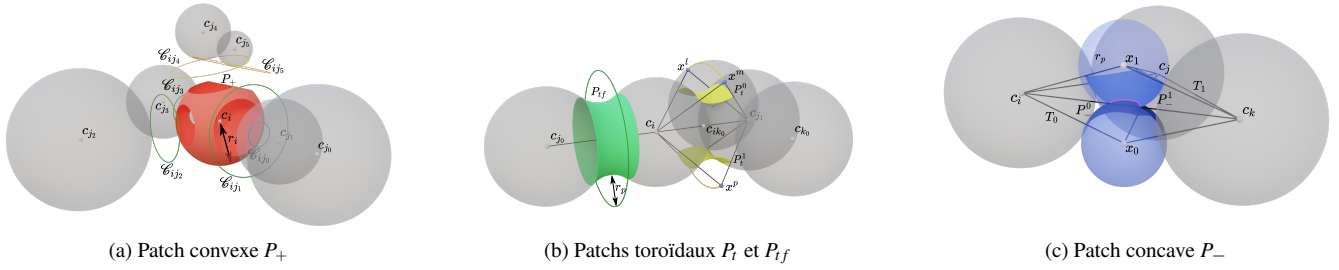


Figure 3: Illustration de la décomposition en patch de la SES ainsi que des éléments structurels de la SAS permettant de calculer leurs SDFs. En vert : cercles complets. En jaune : cercles intersectés.

111 deux sphères et du cercle à leur origine en plus des intersections de
112 la SAS dans le cas de patches toroïdaux partiels P_t .

113 **Patch concave P_- .** Enfin, lorsque le probe est localisé sur une
114 intersection de la SAS, il ne possède plus de degré de liberté. Les
115 patches associés P_- sont donc représentables à partir de la position
116 de l'intersection de la SAS. Cependant, lorsque la sphère du probe
117 dépasse le tétraèdre englobant du patch décrit par la position de
118 l'intersection et celle des trois sphères de la SAS à son origine, il
119 est possible qu'il intersecte un autre patch comme illustré sur la
120 figure 3. Afin de correctement décrire sa surface, il est nécessaire,
121 en plus des informations précédentes, de connaître les intersections
122 de la SAS voisines donnant lieu à des intersections entre les patches.

123 4. Calcul structurel

124 La définition des patches présentée dans la partie précédente permet
125 de calculer la SES de façon complète à partir des primitives de
126 la SAS. Celles-ci représentent donc la cible de notre processus de
127 construction et leurs liens hiérarchiques ordonnent les étapes de
128 notre méthode que nous décrivons dans cette partie.

129 **Calcul et classification des cercles de la SAS.** Les cercles de
130 la SAS sont obtenus à partir des voisinages $\mathcal{N}(a_i)$ de chacun des
131 atomes en sélectionnant ceux dont la distance est inférieure à $r_i +$
132 $r_j + 2r_p$, avec r_i et r_j les rayons des atomes a_i et a_j , et r_p le rayon
133 du probe. Pour cela, nous utilisons une grille accélératrice alliant
134 rapidité de construction et vitesse de requête. Une fois cette donnée
135 obtenue, chaque cercle est classifié en fonction de sa nature selon
136 les types suivants : complet, intersecté et caché [QS17]. Les cercles
137 complets et intersectés vont contribuer aux patches convexes P_+ et
138 respectivement produire des patches toroïdaux P_{tf} et P_t tandis que
139 les patches cachés sont localisés à l'intérieur d'atomes voisins et ne
140 contribuent donc pas à la surface.

141 **Calcul des intersections de la SAS.** Afin de construire les in-
142 tersections de la SAS, il est nécessaire de parcourir l'ensemble
143 des cercles intersectés identifiés à l'étape précédente. Cette étape
144 est difficile puisqu'il est nécessaire de s'assurer que chaque inter-
145 section trouvée n'est pas localisée à l'intérieur d'un autre atome.
146 Pour cela, nous cherchons les atomes présents dans deux voisinages
147 $a_k \in \mathcal{N}(a_i) \cap \mathcal{N}(a_j)$ [QS17]. On assure enfin la visibilité des in-
148 tersections trouvées en vérifiant qu'elles ne sont pas à l'intérieur
149 d'un autre atome du voisinage.

150 **Calcul des segments de la SAS.** Une fois les intersections de
151 la SAS obtenues, nous disposons de toutes les informations néces-
152 saires au calcul des segments de la SAS. La liste des intersections
153 de chaque cercle est triée de façon angulaire, ce qui permet de créer
154 un patch P_t par couple d'intersections [MJK16, QS17].

5. Implémentation GPU

156 Les étapes présentées dans la partie précédente permettent
157 d'obtenir la géométrie de la SES. Cependant, elles intègrent un
158 nombre important d'opérations sur les graphes de voisinage des
159 atomes et des cercles pouvant ralentir de façon importante le
160 traitement, spécifiquement sur GPU, où la cohérence des opéra-
161 tions effectuées en parallèle est particulièrement critique. Dans
162 cette partie, nous présentons donc différents traitements adaptés
163 à ces contraintes. Nous faisons référence aux sous-groupes de fils
164 d'exécutions exécutant la même instruction avec des données dif-
165 férentes sous le terme de warp.

166 **Traitement coopératif des intersections.** Les opérations néces-
167 saires pour traiter une seule intersection entre deux cercles
168 représentent à elles seules une charge importante. Ainsi, nous uti-
169 lions un warp dédié pour le traitement de chacun des cercles in-

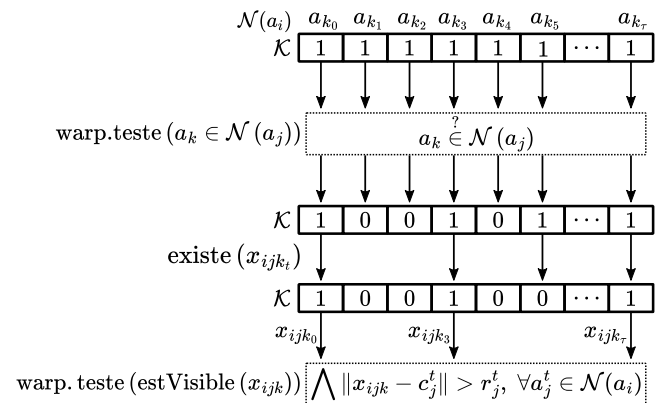


Figure 4: Schéma du traitement coopératif des intersections de la SAS. En pointillés : opérations intrawarp effectuées en parallèle, mais itérativement sur les entrées de \mathcal{K} .

PDB Id	#Atome	Gralka et al. [GBB*19] - ext.		Notre méthode		Notre méthode - ext.	
		Temps (ms)	Mém. (Mo)	Temps (ms)	Mém. (Mo)	Temps (ms)	Mém. (Mo)
101M	1413	4.51	133.55	3.75	4.50	2.51	1.76
7DBB	17733	7.00	1675.43	5.80	56.18	4.49	25.50
7RGD	65008	16.86	6123.51	18.06	214.33	14.64	105.35
3JC8	107640	-	-	14.99	324.03	11.60	147.57
6U42	1358547	-	-	200.79	4287.12	157.23	2162.99
3J3Q	2440800	-	-	700.17	7631.24	324.22	3744.73

Table 1: Comparaison entre notre méthode et l’implémentation publique de Contour-Buildup sur GPU [KGE11] dans le framework Megamol [GBB*19] pour un probe de rayon standard [D’A78] de 1.4 Å. Les temps sont donnés en millisecondes (ms) et représentent une moyenne sur 1000 itérations par cas, tandis que la mémoire maximale utilisée est donnée en mégaoctets (Mo).

170 tersectés de l’atome courant. Comme illustré sur la figure 4, nous
 171 traitons de façons différentes les tâches simples et lourdes du traite-
 172 ment des intersections. Les tâches simples, comme le calcul de la
 173 position de l’intersection, sont traitées à partir d’un seul fil tandis
 174 que les tâches lourdes, comme les tests de visibilité des intersec-
 175 tions, sont traitées de façon séquentielle en utilisant tous les fils du
 176 warp. Ce processus permet ainsi un compromis localisé entre parallélisme
 177 et puissance de calcul. Nous utilisons de plus un masque
 178 binaire afin de conserver les informations sur les tâches au cours du
 179 traitement, ce qui permet une communication efficace entre les fils
 180 d’exécution du warp.

181 **P_- intersectés.** Les patches concaves P_- intersectant leurs tétraè-
 182 dres respectifs sont identifiés pendant leur construction et marqués
 183 comme tels. Ils sont ensuite rassemblés sous la forme d’un sous-
 184 ensemble au début du tableau les stockant ce qui permet d’effectuer
 185 les tests de collision de façon performante et localisée.

186 **Traitement coopératif des segments.** À la suite du calcul des
 187 intersections, nous retrouvons les intersections participant à la SES
 188 de chacun des cercles intersectés de la SAS à l’origine des seg-
 189 ments de la SES. De la même façon que pour les intersections, nous
 190 utilisons ensuite un traitement coopératif. Ainsi nous construisons
 191 d’abord la liste d’intersection de chacun des cercles intersectés que
 192 l’on trie de façon parallèle pour plusieurs cercles à l’intérieur d’un
 193 warp. Cela permet notamment de ne pas souffrir des divergences
 194 induites par les algorithmes de tri séquentiels.

Estimation des besoins mémoires. Il est difficile de prévoir le
 nombre d’intersections $\#\mathcal{X}$ de la SAS [KGE11, SK19]. Nous pro-
 posons de l’estimer à partir du nombre de cercles intersectés $\#\mathcal{I}$:

$$\#\mathcal{X} \leq \#\mathcal{I} \max(\#\mathcal{X}(\mathcal{C}_{ij})) \quad (1)$$

195 Expérimentalement, nous bornons la moyenne maximale du
 196 nombre d’intersections par cercle avec $\max(\#\mathcal{X}(\mathcal{C}_{ij})) = 2$. Celle-
 197 ci n’est pas restrictive en pratique puisque notre implémentation
 198 permet à un cercle unique d’avoir jusqu’à 16 intersections. De
 199 plus, de nombreux cercles intersectés ne produisent finalement
 200 pas d’intersection contribuant à la surface, puisque positionnée à
 201 l’intérieur d’un atome voisin.

202 6. Résultats et performances

Afin d’évaluer les performances de notre méthode, nous com-
 parons sa vitesse d’exécution et sa consommation mémoire à
 celles de la méthode la plus rapide de l’état de l’art selon nos
 tests [KGE11, GBB*19] dans le tableau 1. Ces tests sont effectués
 sur une NVIDIA RTX 2080 8GB et un AMD Ryzen 5 1600. La
 méthode comparée est limitée à la construction de la surface ex-
 térieure, et donc uniquement au calcul des P_{if} et P_- , et sa princi-
 pale limite est sa consommation mémoire [RZK*19, SK19]. Nous
 proposons de plus les résultats de notre méthode adaptée au calcul
 extérieur pour une évaluation dans des conditions similaires.

Nos résultats montrent que notre méthode propose des temps de
 calcul similaires à celle de Krone et al. [KGE11] tout en réduisant
 de façon importante la consommation mémoire. Plus précisément,
 dans le cas de la construction limitée à la surface extérieure, notre
 méthode adaptée consomme 60 fois moins de mémoire. À la dif-
 férence de la méthode comparée, la nôtre permet aussi de calculer
 la surface complète tout en consommant 30 fois moins de mé-
 moire. Cela rend ainsi possible le calcul de la surface complète
 de molécules de plusieurs millions d’atomes sur des GPUs grand
 public.

Comme observé par des travaux précédents [SK19] et visibles
 sur nos résultats, la géométrie de la protéine influe sur les per-
 formances. Ainsi, on peut remarquer que la surface de 3JC8 est
 calculée plus rapidement que celle de 7RDG alors même que son
 nombre d’atomes est plus important. En effet, le nombre de cercles
 intersectés affecte les temps de calcul de manière plus significa-
 tive que le nombre d’atomes. Une molécule dense comporte beau-
 coup de cercles intersectés dont les intersections seront localisées à
 l’intérieur d’atomes voisins et ne participeront pas à la surface.

232 7. Conclusion et travaux futurs

À partir de notre méthode, le calcul de la géométrie complète de
 la SES de grandes molécules est rendu possible avec des perfor-
 mances similaires aux méthodes précédentes. De plus, nous mon-
 trons que nos optimisations permettent une réduction très impor-
 tante de la consommation mémoire du processus. Notre méthode
 pourra donc faciliter l’utilisation des méthodes de calcul analy-
 tique de la SES sur des cartes graphiques plus grand public ainsi
 que le traitement de grandes structures. Cependant, notre méthode

- requiert un recalcul complet en cas de changement de la taille du rayon du probe parfois nécessaire à certaines applications. Nous orientons donc nos travaux pour pallier cette limitation majeure tout en conservant des niveaux de performances importants.
- References**
- [Bru19] BRUCKNER S.: Dynamic visibility-driven molecular surfaces. *Computer Graphics Forum* 38, 2 (May 2019), 317–329. URL: <https://doi.org/10.1111/cgf.13640>, doi: 10.1111/cgf.13640. 2
- [Con83] CONNOLLY M. L.: Analytical molecular surface calculation. *Journal of Applied Crystallography* 16, 5 (oct 1983), 548–558. doi: 10.1107/s0021889883010985. 2
- [D'A78] D'ARRIGO J. S.: Screening of membrane surface charges by divalent cations: an atomic representation. *American Journal of Physiology-Cell Physiology* 235, 3 (sep 1978), C109–C117. doi: 10.1152/ajpcell.1978.235.3.c109. 4
- [EM94] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. *ACM Transactions on Graphics* 13, 1 (jan 1994), 43–72. doi: 10.1145/174462.156635. 2
- [GBB*19] GRALKA P., BECHER M., BRAUN M., FRIESS F., MÜLLER C., RAU T., SCHATZ K., SCHULZ C., KRONE M., REINA G., ERTL T.: MegaMol – a comprehensive prototyping framework for visualizations. *The European Physical Journal Special Topics* 227, 14 (Mar. 2019), 1817–1829. URL: <https://doi.org/10.1140/epjst/e2019-800167-5>, doi:10.1140/epjst/e2019-800167-5. 4
- [HKG*17] HERMOSILLA P., KRONE M., GUALLAR V., VÁZQUEZ P.-P., VINACUA À., ROPINSKI T.: Interactive GPU-based generation of solvent-excluded surfaces. *The Visual Computer* 33, 6-8 (May 2017), 869–881. URL: <https://doi.org/10.1007/s00371-017-1397-2>, doi:10.1007/s00371-017-1397-2. 2
- [JPSK16] JURCIK A., PARULEK J., SOCHOR J., KOZLIKOVÁ B.: Accelerated visualization of transparent molecular surfaces in molecular dynamics. In *2016 IEEE Pacific Visualization Symposium (PacificVis)* (Apr. 2016), IEEE. URL: <https://doi.org/10.1109/pacificvis.2016.7465258>, doi:10.1109/pacificvis.2016.7465258. 2
- [KBE09] KRONE M., BIDMON K., ERTL T.: Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov. 2009), 1391–1398. URL: <https://doi.org/10.1109/tvcg.2009.157>, doi:10.1109/tvcg.2009.157. 2
- [KDE10] KRONE M., DACHSBACHER C., ERTL T.: Parallel computation and interactive visualization of time-varying solvent excluded surfaces. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology* (Aug. 2010), ACM. URL: <https://doi.org/10.1145/1854776.1854840>, doi: 10.1145/1854776.1854840. 2
- [KGE11] KRONE M., GROTTTEL S., ERTL T.: Parallel contour-buildup algorithm for the molecular surface. In *2011 IEEE Symposium on Biological Data Visualization (BioVis)*. (Oct. 2011), IEEE. URL: <https://doi.org/10.1109/biovis.2011.6094043>, doi: 10.1109/biovis.2011.6094043. 2, 4
- [KKF*16] KOZLÍKOVÁ B., KRONE M., FALK M., LINDOW N., BAADEN M., BAUM D., VIOLA I., PARULEK J., HEGE H.-C.: Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum* 36, 8 (Nov. 2016), 178–204. URL: <https://doi.org/10.1111/cgf.13072>, doi:10.1111/cgf.13072. 1
- [MAPV15] MUZIC M. L., AUTIN L., PARULEK J., VIOLA I.: cel-IVIEW: a Tool for Illustrative and Multi-Scale Rendering of Large Biomolecular Datasets. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (2015), Bühler K., Linsen L., John N. W., (Eds.), The Eurographics Association. doi:10.2312/vcbm.20151209. 1
- [MJK16] MANAK M., JIRKOVSKY L., KOLINGEROVA I.: Interactive analysis of connolly surfaces for various probes. *Computer Graphics Forum* 36, 6 (May 2016), 160–172. URL: <https://doi.org/10.1111/cgf.12870>, doi:10.1111/cgf.12870. 3
- [QS16] QUAN C., STAMM B.: Mathematical analysis and calculation of molecular surfaces. *Journal of Computational Physics* 322 (Oct. 2016), 760–782. URL: <https://doi.org/10.1016/j.jcp.2016.07.007>, doi:10.1016/j.jcp.2016.07.007. 2, 5
- [QS17] QUAN C., STAMM B.: Meshing molecular surfaces based on analytical implicit representation. *Journal of Molecular Graphics and Modelling* 71 (Jan. 2017), 200–210. URL: <https://doi.org/10.1016/j.jmkgm.2016.11.008>, doi:10.1016/j.jmkgm.2016.11.008. 2, 3
- [Ric77] RICHARDS F. M.: Areas, volumes, packing and protein structure. *Annual Review of Biophysics and Bioengineering* 6, 1 (jun 1977), 151–176. doi:10.1146/annurev.bb.06.060177.001055. 1, 2
- [RZK*19] RAU T., ZAHN S., KRONE M., REINA G., ERTL T.: Interactive cpu-based ray tracing of solvent excluded surfaces. *Eurographics Workshop on Visual Computing for Biology and Medicine* (2019). URL: <https://diglib.eg.org/handle/10.2312/vcbm20191249>, doi:10.2312/VCBM.20191249. 2, 4, 5
- [SK19] SCHÄFER M., KRONE M.: A massively parallel cuda algorithm to compute and visualize the solvent excluded surface for dynamic molecular data. *Workshop on Molecular Graphics and Visual Analysis of Molecular Data* (2019). URL: <https://diglib.eg.org/handle/10.2312/molva20191094>, doi:10.2312/MOLVA.20191094. 4
- [SOS96] SANNER M. F., OLSON A. J., SPEHNER J.-C.: Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers* 38, 3 (1996), 305–320. doi:10.1002/(SICI)1097-0282(199603)38:3<305::AID-BIP4>3.0.CO;2-Y. 2
- [TA96] TOTROV M., ABAGYAN R.: The contour-buildup algorithm to calculate the analytical molecular surface. *Journal of Structural Biology* 116, 1 (Jan. 1996), 138–143. URL: <https://doi.org/10.1006/jsbi.1996.0022>, doi:10.1006/jsbi.1996.0022. 2

Appendix A: Définitions des fonctions distance signée

Nous décrivons dans cette partie les fonctions distance signée proposées par Quan & Stamm [QS16], utilisées dans ces travaux. Celles-ci permettent d'obtenir la distance entre un point p et la SES en fonction de la SAS.

Patch convexe P_+ Si p n'appartient pas à la SAS, son point le plus proche sur la SES se situe sur un patch convexe P_+ . Sa distance avec la SES est donc obtenue à partir de :

$$d(P_+, p) = \|p - c_i\| - r_i, \quad p \notin s, \quad \forall s \in S_i$$

$$c_i = \arg \min_{\forall c_j \in \mathcal{M}} \|p - c_j\| \quad (2)$$

avec \mathcal{M} la molécule, c_i et r_i les centre et rayon d'un atome a_i , et S_i ses cercles de la SAS. Afin de tester si p appartient à un des cercles de la SAS, nous traduisons ces cercles en secteurs [RZK*19] à partir desquels nous effectuons un test d'inclusion.

Patch toroïdaux P_{tf} et P_t Si p est localisé dans la SAS et plus particulièrement dans le triangle $\triangle(p'c_i c_j)$ décrit par les centres des atomes a_i et a_j , et p' , sa projection sur leur cercle de la SAS

\mathcal{C}_{ij} , alors son point le plus proche sur la SES est localisé sur un patch toroidal dont la distance $d(P_t, p)$ est obtenue avec :

$$\begin{aligned} d(P_t, p) &= -\|p' - p\| + r_p, \quad p \in \Delta(p'c_ic_j) \\ p' &= c_{ij} + r_{ij} \frac{v}{\|v\|} \\ v &= p + ((c_{ij} - p) \cdot n_{ij})n_{ij} - c_{ij} \end{aligned} \quad (3)$$

351 avec c_{ij} et r_{ij} les centre et rayon du cercle \mathcal{C}_{ij} et n_{ij} sa normale
352 orientée de c_i à c_j .

Si le patch associé est un patch toroidal partiel P_t , il est de plus nécessaire de tester si le point appartient au segment visible à partir de :

$$\angle(x^l - c_{ij}, p' - c_{ij}) < \angle(x^l - c_{ij}, x^m - c_{ij}) \quad (4)$$

353 avec x^l et x^m les intersections du patch P_t .

Patch concave P_- Enfin, si p appartient à la SAS et ne correspond à aucun autre type de patch, son point le plus proche est localisé sur un patch concave P_- . Sa distance avec la SES $d(P_-, p)$ est alors obtenue avec :

$$\begin{aligned} d(P_-, p) &= -\|p - x^u\| + r_p \\ x^u &= \underset{\forall x^s \in \mathcal{X}}{\operatorname{arg\,min}} \|p - x^s\| \end{aligned} \quad (5)$$

354 avec \mathcal{X} l'ensemble des intersections de la SAS.