Supplemental of Efficient GPU computation of large protein Solvent-Excluded Surface

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

This document presents supplemental materials for our article entitled "Efficient GPU computation of large protein Solvent-Excluded Surface" for IEEE Transactions on Visualization and Computer Graphics. Notably, we provide additional information supporting the reproduction of our method as well as additional results and illustrations.

CONTENTS

1 COMPUTATION OF CIRCLES, THEIR CLASSIFICATION AND INTERSECTIONS

We recall the important equations involved in the computation of the SES and that we are using in our implementation.

First, the center c_{ij} and radius r_{ij} of a SAS circle $\mathscr{C}_{ij}$ between two atoms a_i and a_j can be obtained with [2]:

$$\begin{aligned} c_{ij} &= c_i + (c_j - c_i) \cdot \frac{(r_i + r_p)^2 - (r_j + r_p)^2 + ||c_i - c_j||^2}{2\,||c_i - c_j||^2} \\ r_{ij} &= \sqrt{(r_i + r_p)^2 - ||c_i - c_{ij}||^2} \end{aligned} \tag{1}$$

A detailed proof of this calculation can be found [3].

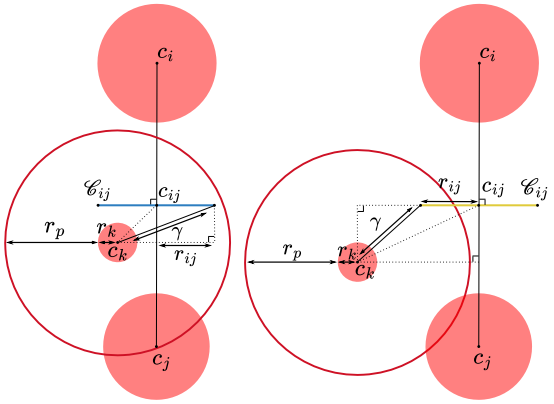


Fig. 1: Circle classification tests. The left schema represents the occlusion test while the right represents the intersection test. In both, γ is the distance that is compared to $r_k + r_p$, the radius of a_k's SAS sphere.

As illustrated in fig. 1, a circle $\mathscr{C}_{ij}$ buried into the SAS sphere of an atom a_k can be identified with [4]:

$$(\sin(\theta)||c_{ij} - c_k|| + r_{ij})^2 + (\cos(\theta)||c_{ij} - c_k||)^2 < (r_k + r_p)^2$$

It can be understood as the length between c_k and its furthest point on $\mathscr{C}_{ij}$, noted γ on the figure, compared to the radius of the SAS sphere. If the first is less than the second, both sides of the circles are occluded by the sphere. In contrast, an intersected circle can be recognized with [4]:

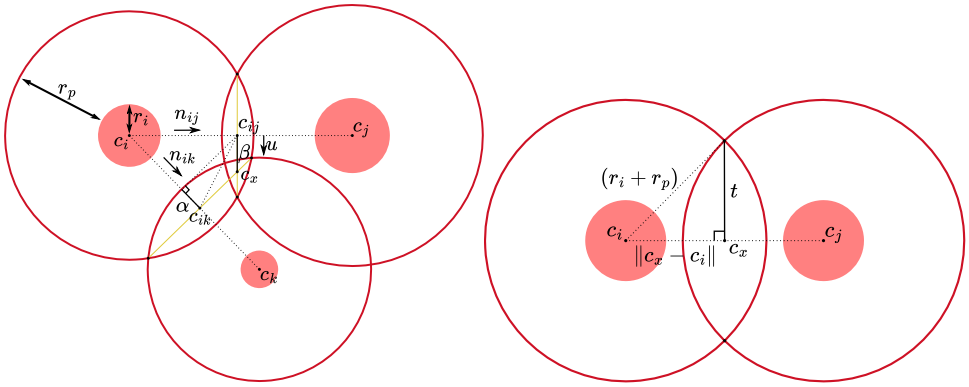
$$(-\sin(\theta)||c_{ij} - c_k|| + r_{ij})^2 + (\cos(\theta)||c_{ij} - c_k||)^2 < (r_k + r_p)^2$$

In this context, and on the right schema of fig. 1, γ is the distance between c_k and its closest point on $\mathscr{C}_{ij}$. If less than the radius of the sphere, we can assert that part of $\mathscr{C}_{ij}$ is intersected.

Fig. 2 illustrates the intersection calculation. It first starts by the computation of c_x, the intersection position on the plane described by c_i, c_j and c_k:

$$c_x = c_{ij} + u\gamma,\ \gamma = \frac{\alpha}{u \cdot n_{ik}}$$
$$\alpha = (c_{ik} - c_{ij}) \cdot n_{ij},\ u = n_{ik} - (n_{ik} \cdot n_{ij})\, n_{ij}$$

where $\gamma = \beta||u||$ is the distance from c_{ij} to c_x in terms of u. The calculation can be understood as the projection of α onto u.



(a) Center c_x (b) Height

Fig. 2: Intersection calculation illustrations. We first compute c_x the intersection position on the plane described by c_i, c_j and c_k. Both intersections $x_{0,1}$ are then located on both sides of this plane at distance t.

Once c_x known, it remains to compute both intersections at the surface of the three spheres x_0 and x_1. As illustrated in fig. 2b, we only have to obtain a normal vector of the plane with $n_{ik} \times n_{ij}$ and compute the height based on one of the spheres' properties:

$$x_{0,1} = c_x \pm (n_{ik} \times n_{ij})\, t$$
$$t = \sqrt{(r_i + r_p)^2 - ||c_x - c_i||^2}$$

2 Additional results

2.1 Scalability

In this section, we give a plot (fig. 3) to compare the scalability of Krone et al. [1]'s method and ours on our test dataset. Note that the former exceeds the 8 GB memory boundary of the GPU used to perform these tests and is not able to compute the surface of the largest proteins of our dataset.

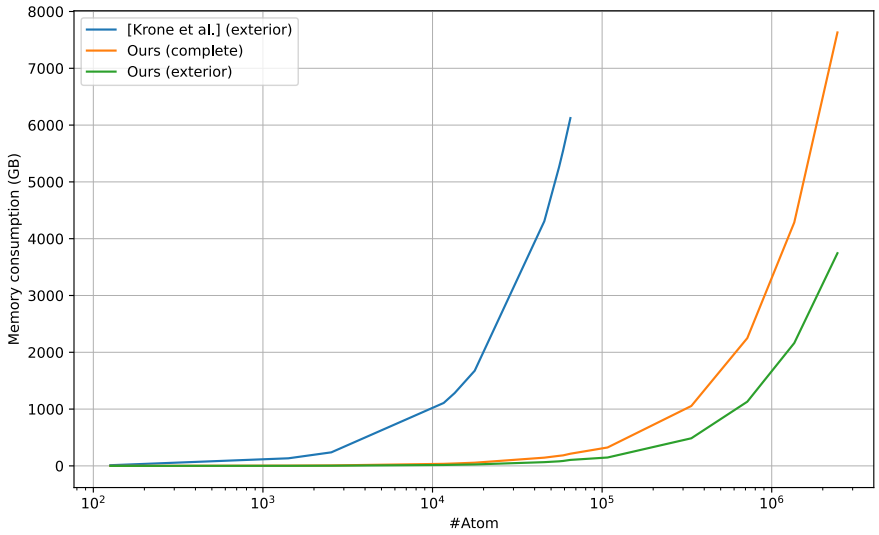


Fig. 3: Memory consumption of the tested methods per atom count on the test dataset performed on a NVIDIA RTX 2080 with 8 GB of memory. Note that the abscissa is representing the atom number in logarithmic scale.

2.2 Experimental results on an NVIDIA Titan RTX

We repeated the experiments with an NVIDIA Titan RTX which offer more memory as well as computation capabilities. These results illustrate the compute scalability of each method and are given in table 1.

PDB Id	#Atom	Krone et al. [1]		Ours		Ours exterior	
		Time (ms)	Mem. (MB)	Time (ms)	Mem. (MB)	Time (ms)	Mem. (MB)
1AGA	126	4.04	11.92	2.58	0.39	1.76	0.15
101M	1413	4.89	133.55	3.00	4.50	2.12	1.76
1VIS	2531	5.23	238.66	2.97	8.01	2.15	3.52
7SC0	11638	6.04	1108.02	3.76	36.17	2.71	16.16
3EAM	13505	5.75	1282.09	4.11	42.69	3.07	19.63
7DBB	17733	6.53	1675.43	4.75	56.18	3.46	25.50
1A8R	45625	8.93	4307.71	7.83	144.92	5.87	65.79
7O0U	55758	9.87	5263.23	8.98	177.25	6.66	80.29
1AON	58870	9.60	5552.49	8.91	185.33	6.59	86.86
7RGD	65008	13.17	6123.51	14.36	214.33	11.06	105.35
3JC8	107640	14.15	10140.00	12.91	324.03	9.81	147.57
7CGO	335722	-	-	37.27	1054.85	28.06	486.03
4V4G	717805	-	-	82.43	2249.47	64.24	1130.35
6U42	1358547	-	-	160.74	4287.12	120.25	2162.99
3J3Q	2440800	-	-	311.31	7631.24	244.36	3744.73

TABLE 1: Experimental results obtained with the same configurations as presented in the paper but with an Intel i9-9900K and an NVIDIA Titan RTX.

2.3 GPU performance of GPU Contour-Buildup [1]

To offer a comparison with the GPU benchmarks given in the main document, we include GPU benchmarks of Megamol's implementation of GPU Contour-Buildup [1] in tab. 4.

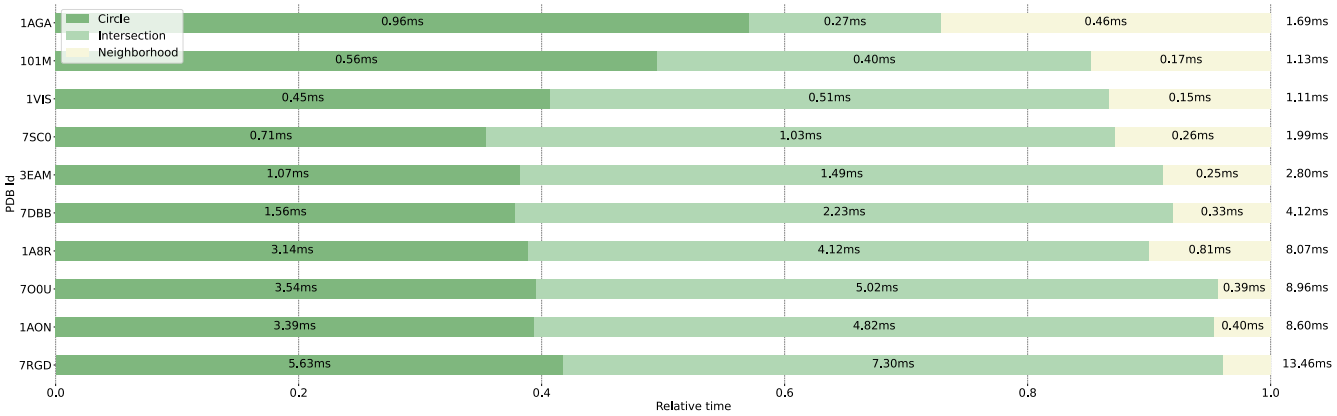


Fig. 4: Detailed GPU benchmarks of the exterior surface computation made with Megamol's implementation of GPU Contour-Buildup [1]. Experiments were performed with a probe radius $r_p = 1.4\,\text{Å}$.

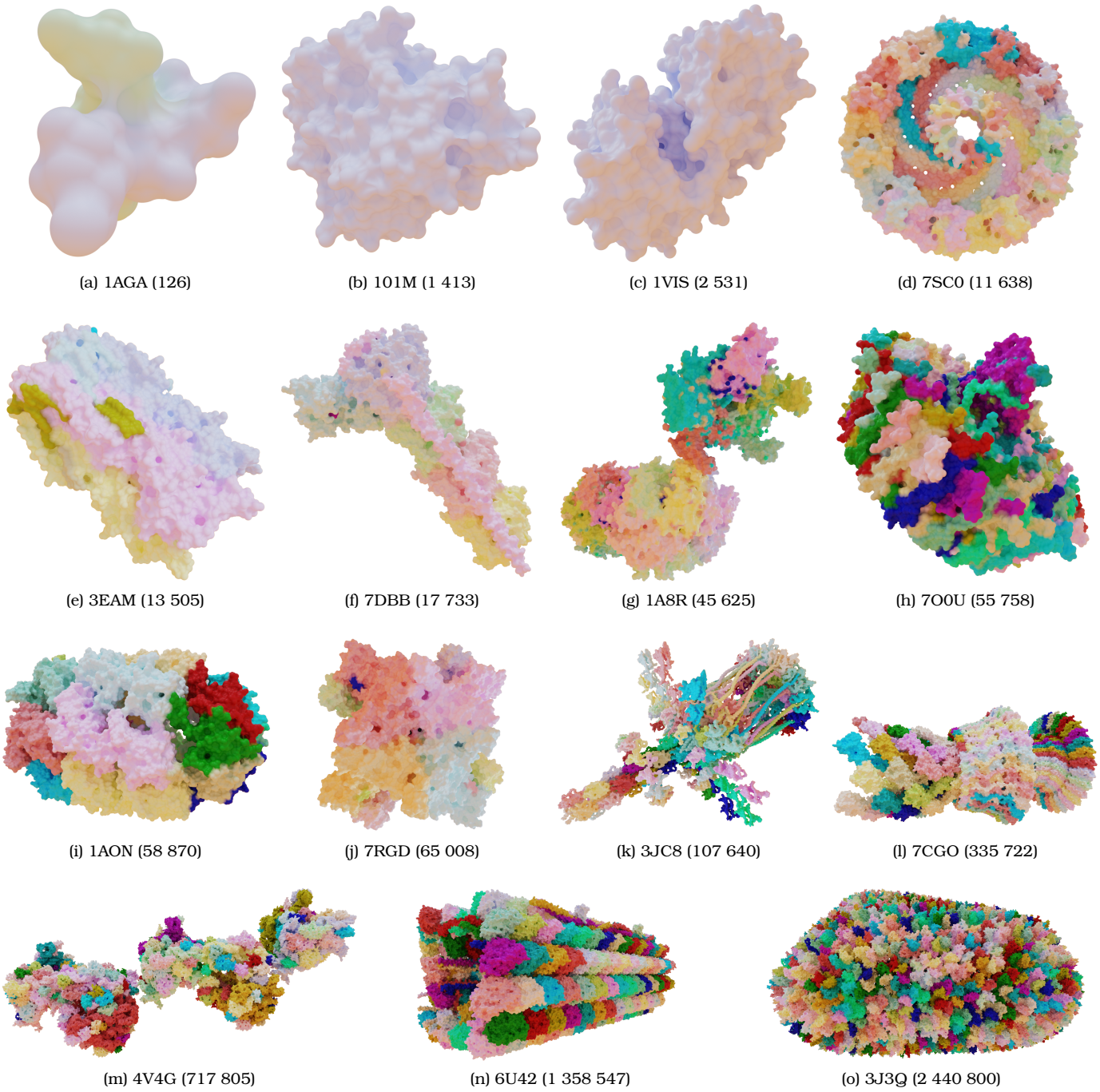
2.4 Rendering benchmarks

Despite being out of the scope of this paper, we provide performance benchmarks made with our real-time rendering engine in tab. 2 with the complete surface visualization as well as the exterior molecular surface only to demonstrate that it is suitable for interactive visualization.

PDB Id	#Atom	Complete surface (ms)	Exterior surface (ms)
1AGA	126	2.08	2.07
101M	1413	3.44	2.78
1VIS	2531	3.78	3.50
7SC0	11638	5.40	4.73
3EAM	13505	6.97	6.31
7DBB	17733	4.89	4.05
1A8R	45625	6.41	5.35
7O0U	55758	10.72	8.41
1AON	58870	9.95	8.59
7RGD	65008	12.74	9.05
3JC8	107640	7.60	7.74
7CGO	335722	20.74	17.57
4V4G	717805	19.31	17.05
6U42	1358547	37.81	36.59
3J3Q	2440800	94.42	32.04

TABLE 2: Rendering benchmarks performed with our rasterization-based engine using the SDF depiction and the exterior-only molecular surface visualization [5]. Experiments were performed with 1000 iterations on randomly sampled points within a sphere centered at the molecule's bounding box center, with a random radius $\in [0,\ 2r_{aabb}]$, a probe radius $r_p = 1.4\,\text{Å}$ and a Lambertian diffuse shading. Given times are then averaged values given in milliseconds.

3 TEST DATASET ILLUSTRATIONS



(a) 1AGA (126) (b) 101M (1 413) (c) 1VIS (2 531) (d) 7SC0 (11 638)

(e) 3EAM (13 505) (f) 7DBB (17 733) (g) 1A8R (45 625) (h) 7O0U (55 758)

(i) 1AON (58 870) (j) 7RGD (65 008) (k) 3JC8 (107 640) (l) 7CGO (335 722)

(m) 4V4G (717 805) (n) 6U42 (1 358 547) (o) 3J3Q (2 440 800)

Fig. 5: Illustration of our test dataset with a standard probe radius $r_p = 1.4$ as well as their respective PDB indices and atoms number.

REFERENCES

[1] M. Krone, S. Grottel, and T. Ertl, "Parallel contour-buildup algorithm for the molecular surface," in *2011 IEEE Symposium on Biological Data Visualization (BioVis)*. IEEE, Oct. 2011. [Online]. Available: https://doi.org/10.1109/biovis.2011.6094043

[2] M. Totrov and R. Abagyan, "The contour-buildup algorithm to calculate the analytical molecular surface," *Journal of Structural Biology*, vol. 116, no. 1, pp. 138–143, Jan. 1996. [Online]. Available: https://doi.org/10.1006/jsbi.1996.0022

[3] E. W. Weisstein, "Circle-circle intersection." MathWorld–A Wolfram Web Resource. [Online]. Available: https://mathworld.wolfram.com/Circle-CircleIntersection.html

[4] C. Quan and B. Stamm, "Meshing molecular surfaces based on analytical implicit representation," *Journal of Molecular Graphics and Modelling*, vol. 71, pp. 200–210, Jan. 2017. [Online]. Available: https://doi.org/10.1016/j.jmgm.2016.11.008

[5] M. Krone, K. Bidmon, and T. Ertl, "Interactive visualization of molecular surface dynamics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1391–1398, nov 2009.