

## Research Article

# Molecular Surface Mesh Generation by Filtering Electron Density Map

Joachim Giard and Benoît Macq

*Communications and Remote Sensing Laboratory, Catholic University of Louvain, B-1348 Louvain-la-Neuve, Belgium*

Correspondence should be addressed to Joachim Giard, joachim.giard@uclouvain.be

Received 25 September 2009; Revised 23 November 2009; Accepted 6 January 2010

Academic Editor: Guo Wei

Copyright © 2010 J. Giard and B. Macq. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bioinformatics applied to macromolecules are now widely spread and in continuous expansion. In this context, representing external molecular surface such as the Van der Waals Surface or the Solvent Excluded Surface can be useful for several applications. We propose a fast and parameterizable algorithm giving good visual quality meshes representing molecular surfaces. It is obtained by isosurfacing a filtered electron density map. The density map is the result of the maximum of Gaussian functions placed around atom centers. This map is filtered by an ideal low-pass filter applied on the Fourier Transform of the density map. Applying the marching cubes algorithm on the inverse transform provides a mesh representation of the molecular surface.

## 1. Introduction

The geometric structure of macromolecules, such as proteins or nucleic acids, is directly related to their function [1–3]. Consequently, studying this structure is of capital importance in the understanding and simulation of numerous life processes. It allows researchers to save a lot of time and money for various applications such as drug design [4, 5] or mutation effect prediction [6, 7]. In this context, working with molecules external surface can be useful, for instance, to predict the geometrical complementarity between two molecules [8] or to visualize them [9]. The prediction of geometric complementarity is one of the keystones of molecular docking [10–13], the modeling of interactions between molecules. The localization of potential binding sites of molecules [14–16] is a frequently used tool for docking and generally requires a good description of the protein surface [17–19]. The estimation of the surface area may also be related to the stability of a particular molecule 3D conformation [20].

At first, the external surface of a molecule has to be defined. Indeed, molecules are made of atoms which have no real surface. The most frequent molecular surface representations are the Van der Waals Surface (VdWS), the Solvent Accessible Surface (SAS), and the Solvent Excluded

Surface (SES) [21, 22]. In the case of the VdWS, the electron clouds around atoms are approximated by rigid spheres with radii corresponding to the Van der Waals (VdW) radii of the atoms. The SAS (resp., SES) is the inner surface of the volume filled by the possible positions of the center (resp., exterior surface) of a ball representing a molecule of solvent, for example, water (see Figure 1).

Efficient tools to represent such surfaces are the polygonal meshes, which are collection of points related by edges and faces that approximate the considered surfaces. A lot of methods have been proposed in the last few years for the generation of a molecular surface meshes.

However, the computational time remains generally high for quality meshes, and it can be a problem when there is a great amount of data to treat. In this paper, we introduce the Filtered Density Map (FDM) algorithm, which is a fast and parameterizable algorithm to generate smooth molecular surface meshes. The generated mesh is the isosurface of frequency filtered electron density map.

This paper is organized as follows. First, some other works related to molecular surface generation are succinctly described in Section 2. Then, the FDM method is described in details in Section 3. And finally, results and comparisons with other methods are presented and discussed in Section 4.

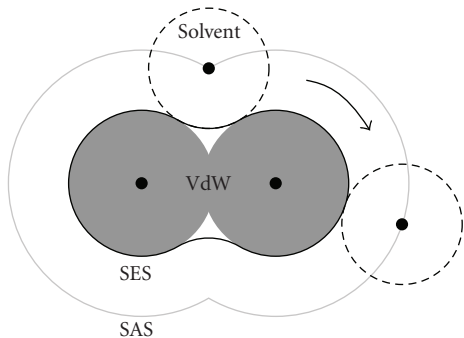


FIGURE 1: A cutaway view of a small molecule. The most frequent molecular surface representations are detailed. Gray discs depict the Van der Waals volumes, the gray outer line depicts the Solvent Accessible Surface (SAS) and the continuous black line depicts the Solvent Excluded Surface (SES). The SAS (resp., SES) is the limit surface for the solvent molecule center (resp., external surface). The solvent molecule is represented as a black dashed circle.

## 2. Related Work

In the last few years, a lot of methods have been developed for the generation of molecular surface meshes. In 1983, Connolly [23] proposed an analytical algorithm in which points were strategically placed around the molecule with a specific analytical role (maximum, minimum or saddle point) depending on the number of atoms present in the neighborhood. In 2003, Bajaj et al. [24] introduced another analytical method based on NURBS that offers the advantage to be parameterizable without recalculation. In 2002, Laug and Borouchaki [25] used a parametric representation of intersecting spheres to create the surface mesh. MSMS, developed by Sanner et al. [26] is based on alpha-shapes [27] of molecules. This algorithm is widely used because it is time efficient. However, the generated mesh is not a manifold and is composed of very irregular triangles. The beta-shapes [28] are a generalization of the alpha-shapes and were used by Ryu et al. [29] in 2007 to design a similar algorithm. Another vertex based method was used by Cheng and Shi [30]. In this method, molecular surfaces are generated with the help of restricted union of balls. Finally, some methods based on volumetric computation exist, such as the one of Zhang et al. [31] in which the solvent accessible surface is seen as the isosurface of Gaussian shaped electron density maps, and the algorithm of Can et al. [32] (the LSMS) which is based on a front propagation from atom center and on level-sets.

Comparisons between the FDM method and the methods mentioned in this section are shown in Section 4.2.

## 3. Method

The FDM method is based on volumetric electron density and a frequency filtering. Each atom is seen as a Gaussian electron cloud, the dimensions of which are depending on the VdW radius. Then, the electron density map is created by taking the local maximum value of these clouds. After a Fourier Transform, it is filtered by an ideal low pass

filter, in order to remove frequencies corresponding to a spatial element smaller than a solvent molecule. Finally, a marching cubes [33] algorithm is used on the inverse Fourier Transform to find an isosurface. A refinement of the final mesh constitutes an optional step of the method. The whole algorithm was implemented in C++ with vtk (Visual ToolKit) (<http://www.vtk.org/>).

3.1. *Electron Density Map.* A Gaussian function is constructed around each atom. The value of this function at a point  $i$  for an atom  $a$  is:

$$G_a(i) = te^{-(r_a^2 - \|ai\|^2/r^2)}, \quad (1)$$

where  $t$  is a threshold parameter,  $r$  is a radius parameter,  $\|ai\|$  is the Euclidean distance between the center of  $a$  and the point  $i$ ,  $r_a$  is the VdW radius of the atom  $a$ . So, the isosurface for the threshold  $t$  is the VdW sphere because if  $i$  is located on the VdWS of  $a$ ,  $\|ai\| = r_a$  and  $G_a(i) = t$ . In this work,  $r$  is set to 3 Å because this value is suitable for an ideal low-pass filter (see Section 3.2).

For the implementation, the three-dimensional space is divided into voxels. The spacing ( $T_e$ ) between voxels is an adaptable parameter. The more  $T_e$  is small, the more the surface approximation is fine.

The density map of the whole molecule for a point in the space is defined as the maximal value of all the Gaussian functions at this point. The maximum of the Gaussian functions is chosen instead of the summation because it is not possible to evaluate the SES using the isosurface of a summation of Gaussian functions. It can be shown by the following counterexample, in which the Gaussian affected to the atoms  $a$  and  $b$  must have contradictory properties depending on the situation.

In the first situation, the space between  $a$  and  $b$  is just small enough to block the way to a solvent molecule (see Figure 2(a)). The other atoms are considered to be too far to have an influence. Thus, the SES has a concave shape at this place and the value of the density map at the “center” of the concavity  $c_1$  must be influenced by the fields of  $a$  and  $b$ :  $G_a(c_1) > 0$ ,  $G_b(c_1) > 0$ . We can state that  $\|ac_1\| > r_a + r_s$  because  $s$ , the center of the solvent molecule, can be very close to the  $ab$  axis.

In the second situation,  $\|ab\| < r_a + r_s/2$ , which is often the case for covalent bonds. Let  $c_2$  be a point belonging both to the SES and to the VdWS of  $b$  with the necessary condition  $\|bc_2\| = r_b$  (see Figure 2(b)). The point  $c_2$  should not be influenced by the field of  $a$ , so  $G_a(c_2) = 0$ . If  $\|ac_2\| > \|ac_1\|$ , this condition is in contradiction with  $G_a(c_1) > 0$ , because the Gaussian function is strictly decreasing in the positive domain. Using the Al-Kashi theorem, we know that

$$\|ac_2\|^2 = r_b^2 + \|ab\|^2 - 2r_b\|ab\| \cos \beta,$$

$$\|ac_1\|^2 > (r_a + r_s)^2 = (r_b + r_s)^2 + \|ab\|^2 - 2(r_b + r_s)\|ab\| \cos \beta, \quad (2)$$

where  $\beta$  is the  $\widehat{abf}$  angle. Thus,  $\|ac_1\| > \|ac_2\|$  if

$$r_s^2 + 2r_b r_s - 2r_s \|ab\| \cos \beta > 0, \quad (3)$$

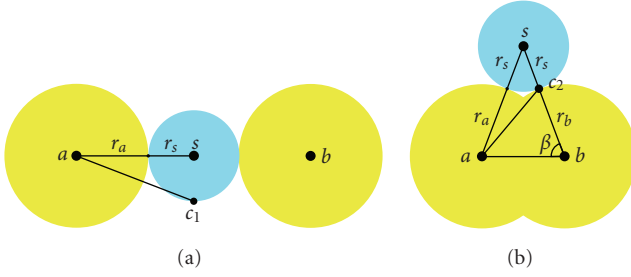


FIGURE 2: Two situations in which the Gaussian functions would have contradictory properties. (a) Two atoms of the molecule,  $a$  and  $b$ , are just close enough to block the way to a solvent molecule  $s$ . The point  $c_1$  is on the SES and must be influenced by the field generated by  $a$ ,  $G_a$ . (b)  $a$  and  $b$  are close. The point  $c_2$  is on the SES but also on the VdWS. So, it should not be influenced by  $G_a$ . It is possible to show that in some (frequent) configurations,  $\|ac_1\| > \|ac_2\|$ .

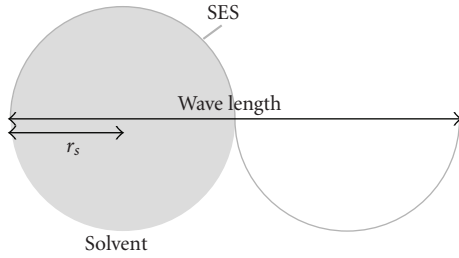


FIGURE 3: Minimal wavelength allowed on the SES. It corresponds to four times the solvent radius  $r_s$  and determines the cutoff frequency for the filtering.

what is verified by the hypothesis:  $\|ab\| < r_b + r_s/2$  because  $\cos\beta \leq 1$ .

In order to avoid interferences, the maximum is preferred to the summation of Gaussian functions. Isosurfacing this density map returns the VdWS. This surface is not smooth and in order to compute the SES, the density map must first be filtered.

**3.2. Fourier Transform and Filtering.** The Fourier Transform of this electron density map is computed using the FFT algorithm [34]. The frequency representation of the function is filtered by an ideal low pass filter in order to eliminate frequencies corresponding to elements smaller than a solvent molecule, for example, inflexion points between two VdW spheres. The cutoff frequency is  $f_c = 1/4r_s$ , where  $r_s$  is the radius of the sphere approximating the solvent molecule (typically  $1.4 \text{ \AA}$  for water). The wavelength must be four times longer than  $r_s$  because a molecule solvent diameter has to fit in a half wavelength (see Figure 3).

Gaussian functions are preferred to balls in the spatial domain because an ideal low-pass filter makes the Gibbs phenomenon appear on sharp edges. An ideal filter is used because the cutoff frequency is exactly known and because it is numerically possible. An ideal low-pass filter in the frequency domain is equivalent to a convolution product with a sinc function in the space domain. Let  $X = (x, y, z)$  be

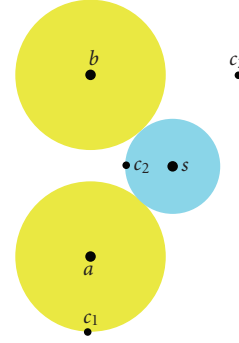


FIGURE 4: In this example, the atoms  $a$  and  $b$  are  $5 \text{ \AA}$  far from each other and have both a VdW radius of  $1.8 \text{ \AA}$ . The radius of the solvent molecule  $s$  is  $1.4 \text{ \AA}$ . The conditions to verify for the density map before ( $\mathcal{M}$ ) and after ( $\tilde{\mathcal{M}}$ ) filtering are:  $\mathcal{M}(c_1) = \tilde{\mathcal{M}}(c_1) = t$ ,  $\mathcal{M}(c_2) < \tilde{\mathcal{M}}(c_2) = t$ , and  $\mathcal{M}(c_3) < t$  and  $\tilde{\mathcal{M}}(c_3) < t$ , with  $t$ , the threshold value for the isosurfacing.

the space variable in  $\mathbb{R}^3$  and  $\mathcal{M}(X)$  the initial electron density map. Then, the filtered density map is:

$$\tilde{\mathcal{M}}(X) = (\mathcal{M}(\bullet) * 2f_c \text{sinc}(2f_c \bullet))(X), \quad (4)$$

where  $\bullet$  represents the variable. The parameter  $r$  of the Gaussian functions in (1) is related to the width of the function. To keep the isosurface at the same place, a wider function has a smaller maximum. If this maximum is too high, that is, if  $r$  is too small, the secondary ripples of the sinc function take too much importance when they are in phase with this maximum. It makes oscillations appear in the final density map, which can lead to the apparition of unwanted surfaces after isosurfacing. Simulations with several 1D and 2D functions were performed to verify the effect of  $r$ . The three main conditions to verify are the following:

$$\begin{aligned} \tilde{\mathcal{M}}(X_{c_1}) &= t && \text{for } c_1 \text{ lying on the SES and on the VdWS,} \\ \tilde{\mathcal{M}}(X_{c_2}) &= t && \text{for } c_2 \text{ lying on the SES but not on the VdWS,} \\ \tilde{\mathcal{M}}(X_{c_3}) &< t && \text{for } c_3 \text{ lying outside the molecule.} \end{aligned} \quad (5)$$

Here is a 2D example:

$$\mathcal{M}(x) = \max(G_a(x), G_b(x)), \quad (6)$$

with the atom  $a$  centered in  $(x_a, y_a) = (0, -2.5)$  and the atom  $b$  in  $(x_b, y_b) = (0, 2.5)$ , the threshold  $t = 1$ , and  $r_a = r_b = 1.8$  (Figure 4). The values of the density maps  $\mathcal{M}(X)$  and  $\tilde{\mathcal{M}}(X)$  for  $c_1 = (0, x_a - r_a)$ ,  $c_2 = (\sqrt{x_a^2 + (r_a + r_s)^2} - r_s, 0)$ , and  $c_3 = (x_3, y_3)$  are plotted as a function of  $r$  in Figure 5.  $(x_3, y_3)$  is the position of the maximal value of the density map outside the "molecule" for  $r = 1$ . When the parameter  $r = 3$ , conditions (5) are verified and the Gaussian functions are not too wide, what leads to shorter execution times. The 2D density maps  $\mathcal{M}(X)$  and  $\tilde{\mathcal{M}}(X)$  with  $r = 3$  as well as  $\tilde{\mathcal{M}}(X)$  with  $r = 1$  are shown in Figure 6. The isocontours, representing the VdWS

TABLE 1: Computation times (in s) for different methods. <sup>(b)</sup> from [32].)

PDB code	No. atoms	FDM <sub>1.9r<sub>s</sub>,0</sub>	FDM <sub>4/3r<sub>s</sub>,1</sub>	LSMS	PyMol	Swiss-PDBV	Chimera	MSMS
1A8R	26400	0.65	2.61	5.56	10.52	6.38	16.36 <sup>(b)</sup>	0.95
1H2I	32318	0.72	1.83	6.50	11.37	5.25	40.04 <sup>(b)</sup>	3.03
1GTP	34740	0.51	2.20	6.98	13.15	4.75	67.04 <sup>(b)</sup>	9.02
1FKA	34977	0.75	2.77	7.89	26.29	7.36	77.25 <sup>(b)</sup>	4.50
1GT7	42700	0.95	2.43	7.32	16.10	6.50	54.39 <sup>(b)</sup>	3.32
1GAV	43335	0.55	2.54	7.05	28.86	7.71	78.35 <sup>(b)</sup>	4.22
1G3I	45528	0.54	2.85	8.18	19.45	6.21	—	7.67
1PMA	45892	0.40	1.97	8.23	18.67	6.72	—	12.90
1FJG	51995	0.71	2.88	8.01	25.18	8.05	—	15.11
1AON	58870	0.62	2.64	8.83	26.36	8.91	—	10.87
1J0B	60144	0.69	3.07	6.87	32.66	7.92	—	5.61
1OTZ	68620	0.67	2.28	8.46	30.14	9.56	—	9.03
1IR2	77088	0.65	2.88	7.09	29.31	9.55	93.87 <sup>(b)</sup>	9.49

TABLE 2: Computation times (in s) for different methods. <sup>(a)</sup> from [30].)

PDB code	No. atoms	≈ No. triangles	FDM <sub>r<sub>s</sub>/2,1</sub>	Cheng	MSMS
200D	232	65 k	0.55	1.35 <sup>(a)</sup>	0.33
1FG1	873	100 k	0.85	2.41 <sup>(a)</sup>	0.65
3EBZ	1651	200 k	1.37	15.43 <sup>(a)</sup>	0.97

TABLE 3: RMSD and percentage of big differences with reference surfaces.

Spacing (Å)	RMSD <sub>w</sub> (Å)	$\  \cdot \  > 1.9r_s$	$\  \cdot \  > r_s$	$\  \cdot \  > r_s/4$
1.9r <sub>s</sub> = 2.66	0.78	<b>0.41%</b>	6.89%	67.04%
r <sub>s</sub> = 1.4	0.26	0%	<b>0%</b>	20.20%
r <sub>s</sub> /4 = 0.35	0.20	0%	0%	<b>8.71%</b>

or the SES, are depicted in white and we can see the artifacts appearing for too small values of  $r$ .

It is important to notice that if the spacing ( $T_e$ ) for the spatial sampling is too large, there would be no filtering. Normally, the sampling frequency,  $f_e = 1/T_e$ , has to verify the Nyquist-Shannon theorem:  $f_e > 2f_{\max}$ , where  $f_{\max}$  is the higher frequency with a nonzero coefficient in the original signal. However, for this application the errors resulting from a subsampling are not too important and the sampling frequency is chosen such that  $f_e > 2f_c$ , that is,  $T_e < 2r_s$ . In this situation, the filtering is always possible.

**3.3. Isosurfacing.** The final triangular mesh is an approximation of the isosurface of the filtered electron density map. The most popular technique to extract an isosurface from a 3D image is the marching cubes algorithm [33]. In this algorithm, the voxels are screened by group of eight sharing a same point. Mesh vertices, faces, and edges are added depending on the value of these eight voxels. There are 256 ( $2^8$ ) possibilities that can be reduced to 15 situations thanks to symmetries and complementarities.

**3.4. Refinement.** The visual appearance of the final mesh can be improved by magnifying the number of vertices. The number of vertices is increased using a smooth interpolation scheme such as the piecewise smooth surface reconstruction of Hoppe et al. [35], or the algorithm based on the butterfly scheme proposed by Zorin et al. [36].

## 4. Results and Discussion

Some numerical results pointing out advantages and drawbacks of the FDM are shown in this section. The main characteristics to be observed are the computation time and the quality of the generated mesh. The section is divided into three parts: the analysis of the effects of the different parameters of the FDM, the results of computation time comparisons with other existing methods, and a quality measurement of the generated meshes.

**4.1. Parameters.** There are three main parameters modifiable by the user. First, the spatial spacing  $T_e$ , that is, the distance between two neighbor voxels center, which determines the total number of voxels. Second, the cutoff frequency  $f_c$ , which determines the smoothness of the final mesh. And third, the refinement rate  $k_r$ , that is, the number of new points in a triangle for the final mesh magnification. In this section, the effect of these parameters on the visual quality, on the computation time and on the memory space, are discussed.

**4.1.1. Spatial Spacing.** With a small spatial spacing, it is possible to represent fine details. However, it drastically increases

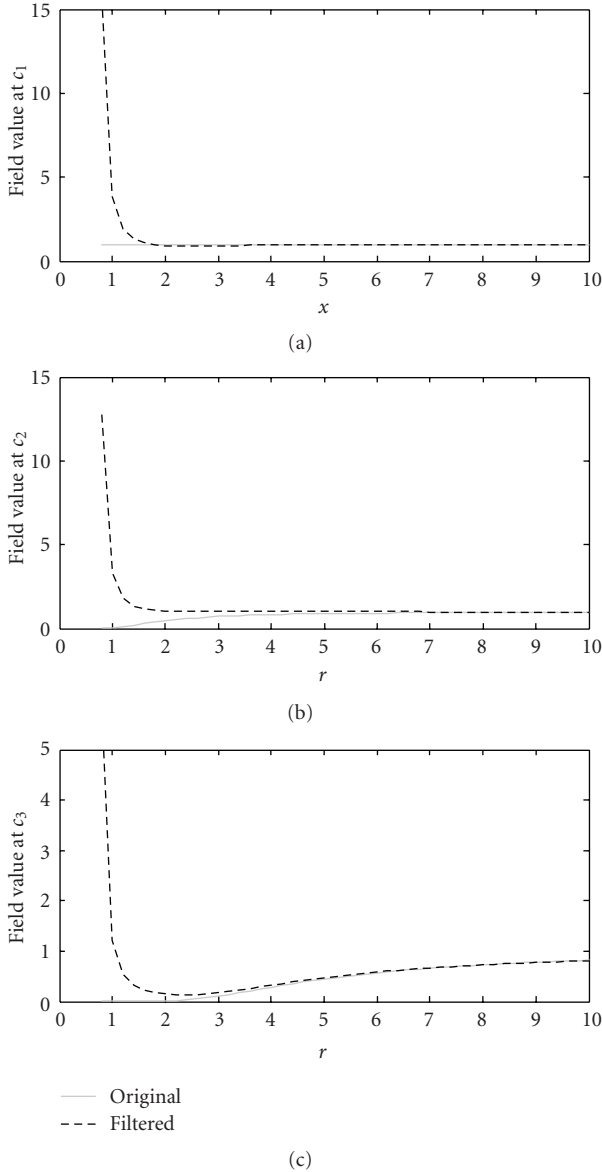


FIGURE 5: Value of the density map before (continuous gray line) and after (dashed black line) filtering at a point belonging to both the SES and the VdWS,  $c_1$  (a) a point belonging to the SES but not the VdWS,  $c_2$  (b) and a point outside the molecule,  $c_3$  (c) (see Figure 4).  $t = 1$ , so, when  $r = 3$ , conditions (5) are satisfied.

the memory space needed as well as the computation time. Indeed, reducing  $T_e$  by a factor  $\alpha$  increases the number of voxels by  $\alpha^3$ . The parts of the method depending on the number of voxels are the creation of the density map (time:  $\mathcal{O}(n_v)$  and space:  $\mathcal{O}(n_v)$ , with  $n_v$  the number of voxels) and the Fast Fourier Transform (time:  $\mathcal{O}(n_v \log n_v)$  and space:  $\mathcal{O}(n_v)$ ). A visual comparison between meshes generated with  $T_e = 1.9r_s$ ,  $T_e = r_s$  and  $T_e = r_s/3$  is shown in Figure 7. In these examples,  $f_c = 1/4r_s$  and  $k_r = 0$ , that is, the meshes represent the SES without final refinement.

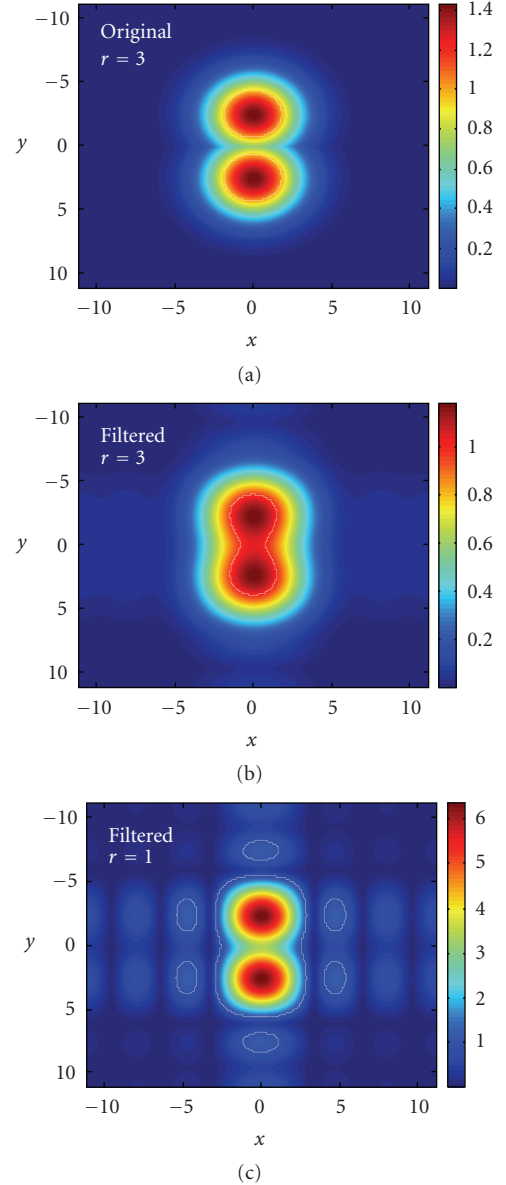


FIGURE 6: Value of the density map for the example of Figure 4.  $x$  and  $y$  are the two spatial dimensions ( $\text{\AA}$ ) and the colors represent the value of the density map before filtering and with  $r = 3$  (a) after filtering and with  $r = 3$  (b) and after filtering with  $r = 1$  (c). The contour for  $t = 1$  is depicted with a white line. It represents the VdWS (a), the SES (b), and the SES with artifacts (c).

**4.1.2. Cutoff Frequency.** In order to generate a mesh representing the SES,  $f_c$  is set to  $1/4r_s$  (see Section 3.2). However, depending on the application, the surface could be other than the SES. For instance, if  $f_c > f_e/2$ , there is no actual filtering, and so, the generated mesh represents the VdWS. On the other hand, to obtain a smooth approximation of the molecule shape,  $f_c$  can be reduced. It is equivalent to consider a bigger solvent molecule. Changing this parameter does not have any effect neither on the computation time nor on the memory space needed. A visual comparison between meshes generated with  $f_c > f_e/2$  (VdWS),  $f_c = 1/4r_s$ , and

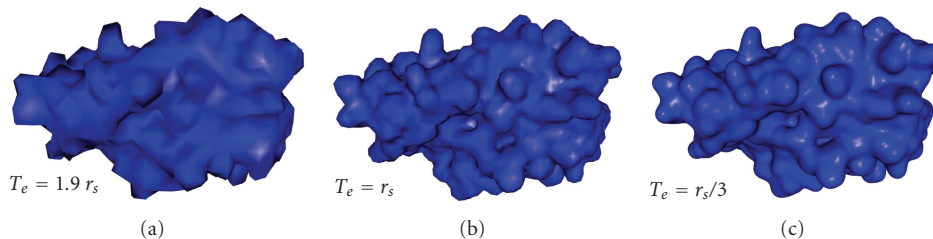


FIGURE 7: Meshes generated from electron density map at a spatial spacing of  $T_e = 1.9r_s$  (a),  $T_e = r_s$  (b), and  $T_e = r_s/3$  (c) (PDB code: 3EBZ).

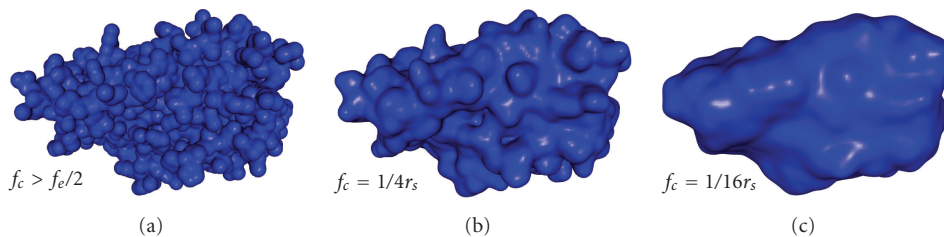


FIGURE 8: Meshes generated from electron density map filtered at a cutoff frequency of  $f_c > f_e/2$  (a) to give the VdWS,  $f_c = 1/4r_s$  (b) to give the SES, and  $f_c = 1/16r_s$  (c) to give an approximation of the general shape. (PDB code: 3EBZ).

$f_c = 1/16r_s$  (shape approximation) is shown in Figure 8. In these examples,  $T_e = r_s/3$  and  $k_r = 0$ , that is, a solvent molecule diameter takes 6 voxels and there is no final refinement.

**4.1.3. Refinement Factor.** The final mesh refinement gives foremost an esthetic advantage. The memory space needed does not increase a lot because the number of voxels remains the same. Only the size of the mesh changes and this is negligible in comparison with the space needed by the voxels representation. The computation time slightly increases but, when  $k_r = 1$  (which gives a good visual improving), this is negligible in comparison with the voxels operations computation time. A visual comparison between meshes generated with  $k_r = 0$ ,  $k_r = 1$  and  $k_r = 2$  is shown in Figure 9. In these examples,  $T_e = r_s$  and  $f_c = 1/4r_s$ , that is, a solvent molecule diameter takes 2 voxels and the meshes represent the SES.

**4.2. Time Comparisons.** In this section, computation times are compared between the FDM algorithm and algorithms found in the literature for equivalent qualities. When available, the algorithms were run on the same computer, when not, the computation times were the ones announced in the original paper. Can et al. made a comparison of their method computation time with three molecular visualization tools: UCSF Chimera [37], Swiss-PDBViewer [38], and PyMol [9]. We added MSMS [26] to this set of methods. These programs, as well as the one of Can (LSMS), are available for free, so, the computation time could be measured on the same computer than for our method, except for Chimera that was not supported by the system. Thus, the computation times mentioned here for Chimera are the one announced in the paper of Can et al. [32]. For the LSMS method, the grid size is set to  $256 \times 256 \times 256$ . In this condition,  $T_e \simeq$

1 with the tested molecules. Other programs are run with defaults settings, that is,  $T_e \simeq 1$ . Two tests are made for the FDM method. In the first one, the parameters are set to  $T_e = 1.9r_s$  and  $k_r = 0$  to be as fast as possible while keeping a correct solution. In the second one, the parameters are set to  $T_e = 4/3r_s$  and  $k_r = 1$  which gives a correct mesh with a good appearance (see Figure 10). The computation times are shown in Table 1. The computation times reported only include the mesh generation time, that is, it does not take the loading time into account. In addition, for three molecules, computation times for the method of Cheng and Shi [30] are reported from their paper. The computation times comparison is shown in Table 2. All the tests were performed on a AMD Athlon(tm) 64 X2 Dual Core Processor 3800+, 2 gigabytes RAM. The computers were more or less equivalent in the cited papers.

It appears in Tables 1 and 2 that all the molecule surfaces in this data set are computed faster with the FDM than with any other method and for different values of the parameters.

**4.3. Quality Results.** In order to validate the quality of the results, different generated surfaces (SESs) were compared with references surfaces. These reference surfaces were generated by isosurfacing a field composed of a union of VdW balls at good resolution (spatial spacing of  $r_s/8$ ) after morphological closing with a structuring element of the size of the solvent molecule. This approach, similar to [32], directly follows the definition of the SES [22], because morphologically close this volume is equivalent to make a solvent molecule roll on the VdW balls and to consider inaccessible parts to be inside the molecule. The molecules tested were 200D, 1FG1, and 3EBZ, because they are small enough to generate a good reference surface with the available memory space. The mean weighted root

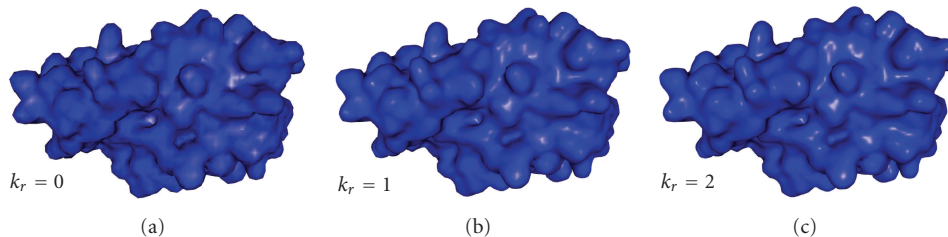


FIGURE 9: Meshes refined with a factor of  $k_r = 0$  (a),  $k_r = 1$  (b), and  $k_r = 2$  (c) (PDB code: 3EBZ).

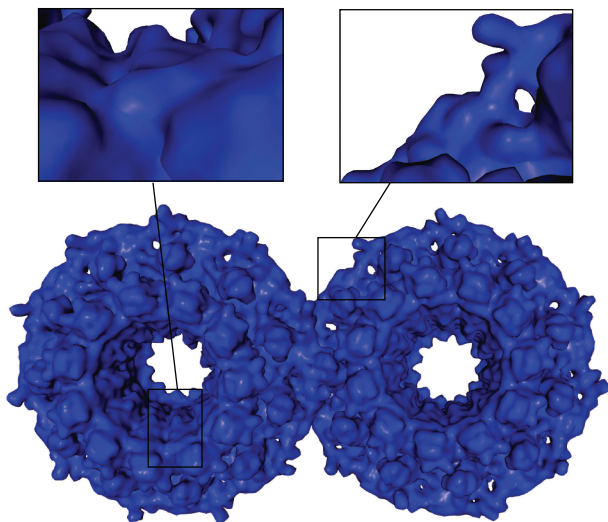


FIGURE 10: Mesh of the SES of a molecule with  $T_c = 4/3r_s$  and  $k_r = 1$ . (PDB Code: 1H2I).

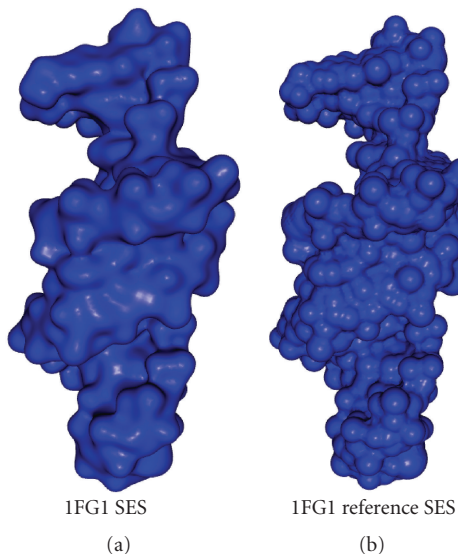


FIGURE 11: Visual comparison between the 1FG1 SES generated with the FDM algorithm with a spacing of  $r_s/4$  and the reference SES generated with a spacing of  $r_s/8$ .

mean square deviation (RMSD) for three spatial spacing is reported in Table 3. The weighted RMSD is

$$\text{RMSD}_w = \sqrt{\frac{\sum_{n=1}^N \|p_n p'_n\|^2 s_n}{S}}, \quad (7)$$

where  $N$  is the number of vertices in the reference mesh,  $p_n$  is a vertex of this mesh,  $p'_n$  is the closest point on the other mesh (not necessary a vertex, it can be on an edge or on a face),  $s_n$  is the mean area of the faces  $p_n$  belongs to, and  $S$  is the total surface area. The percentages of the surface for which  $\|p_n p'_n\|$  is greater than the spacings are shown in the right-hand-side columns of Table 3. These error indexes are not completely correct because the reference surfaces is not a ground truth. However, it shows that the FDM algorithm can provide a surface with a quality comparable to robust methods.

A visual comparison between the SES of 1FG1 computed with the FDM algorithm and the reference SES is shown in Figure 11.

## 5. Conclusion

In this paper, we introduced an algorithm to compute molecular surface meshes (the FDM algorithm). It is constructed as an isosurface of a filtered electron density map (FDM). This algorithm is faster than other algorithm tested in equivalent conditions. It is slower than the MSMS algorithm for small molecules (<30000 atoms) but it returns a smooth manifold surface, which is not the case with MSMS. It makes possible to compute a precise representation of the surface with a limited number of voxels, so that the computation time and the memory space needed are reduced. Moreover, it is parameterizable on the spatial resolution, the refinement of the final mesh, and the size of the solvent molecule. Thus, the spatial resolution can be improved for a finer result but with an important computation time increase. Similarly, a smoother result can be obtained with a final refinement with a small influence on the computation time but with less precise results than reducing the spacing. Finally, the solvent molecule size can be chosen without influence on the computation time.

The refinement could be improved to be specific to molecular surface. It would enable coarse meshes to be generated rapidly and to be improved by a priori knowledge about local geometry of molecule surfaces, such that the

curvature deduced from the closest atom radius. In future works, this algorithm will be used in surface-based method to detect protein hot spots [19].

## Acknowledgment

This research is funded by the NANOTIC/Tsarine project of the Région Wallonne (Belgium).

## References

- [1] J. S. Fetrow and J. Skolnick, "Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases," *Journal of Molecular Biology*, vol. 281, no. 5, pp. 949–968, 1998.
- [2] C. Zhang and S.-H. Kim, "Overview of structural genomics: from structure to function," *Current Opinion in Chemical Biology*, vol. 7, no. 1, pp. 28–32, 2003.
- [3] J. D. Watson, R. A. Laskowski, and J. M. Thornton, "Predicting protein function from sequence and structural data," *Current Opinion in Structural Biology*, vol. 15, no. 3, pp. 275–284, 2005.
- [4] I. D. Kuntz, "Structure-based strategies for drug design and discovery," *Science*, vol. 257, no. 5073, pp. 1078–1082, 1992.
- [5] M. Congreve, C. Murray, and T. Blundell, "Keynote review: structural biology and drug discovery," *Drug Discovery Today*, vol. 10, no. 13, pp. 895–907, 2005.
- [6] M. Prevost, S. J. Wodak, B. Tidor, and M. Karplus, "Contribution of the hydrophobic effect to protein stability: analysis based on simulations of the ile-96- ala mutation in barnase," *Proceedings of the National Academy of Sciences*, vol. 88, no. 23, pp. 10880–10884, 1991.
- [7] Z. W. Cao, L. Y. Han, C. J. Zheng, et al., "Computer prediction of drug resistance mutations in proteins," *Drug Discovery Today*, vol. 10, no. 7, pp. 521–529, 2005.
- [8] S. Bespamyatnikh, V. Choi, H. Edelsbrunner, and J. Rudolph, *Accurate Protein Docking by Shape Complementarity Alone*, Duke University, Durham, NC, USA, 2004.
- [9] W. DeLano, *The PyMOL Molecular Graphics System*, DeLano Scientific, San Carlos, Calif, USA, 2002.
- [10] D. Fischer, S. L. Lin, H. L. Wolfson, and R. Nussinov, "A geometry-based suite of molecular docking processes," *Journal of Molecular Biology*, vol. 248, no. 2, pp. 459–477, 1995.
- [11] H. A. Gabb, R. M. Jackson, and M. J. E. Sternberg, "Modelling protein docking using shape complementarity, electrostatics and biochemical information," *Journal of Molecular Biology*, vol. 272, no. 1, pp. 106–120, 1997.
- [12] N. Brooijmans and I. D. Kuntz, "Molecular recognition and docking algorithms," *Annual Review of Biophysics and Biomolecular Structure*, vol. 32, pp. 335–373, 2003.
- [13] Y. Wang, P. K. Agarwal, P. Brown, H. Edelsbrunner, and J. Rudolph, "Coarse and reliable geometric alignment for protein docking," in *Proceedings of the Pacific Symposium on Biocomputing*, pp. 65–75, 2005.
- [14] G. J. Bartlett, C. T. Porter, N. Borkakoti, and J. M. Thornton, "Analysis of catalytic residues in enzyme active sites," *Journal of Molecular Biology*, vol. 324, no. 1, pp. 105–121, 2002.
- [15] P. Bate and J. Warwicker, "Enzyme/non-enzyme discrimination and prediction of enzyme active site location using charge-based methods," *Journal of Molecular Biology*, vol. 340, no. 2, pp. 263–276, 2004.
- [16] R. Chakrabarti, A. M. Klibanov, and R. A. Friesner, "Computational prediction of native protein ligand-binding and enzyme active site sequences," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 29, pp. 10153–10158, 2005.
- [17] J. R. Bradford, C. J. Needham, A. J. Bulpitt, and D. R. Westhead, "Insights into protein-protein interfaces using a Bayesian network prediction method," *Journal of Molecular Biology*, vol. 362, no. 2, pp. 365–386, 2006.
- [18] F. K. Pettit, E. Bare, A. Tsai, and J. U. Bowie, "HotPatch: a statistical approach to finding biologically relevant features on protein surfaces," *Journal of Molecular Biology*, vol. 369, no. 3, pp. 863–879, 2007.
- [19] J. Giard, J. Ambroise, J. L. Gala, and B. Macq, "Regression applied to protein binding site prediction and comparison with classification," *BMC Bioinformatics*, vol. 10, no. 1, p. 276, 2009.
- [20] T. Arakawa and S. N. Timasheff, "Stabilization of protein structure by sugars," *Biochemistry*, vol. 21, no. 25, pp. 6536–6544, 1982.
- [21] N. Max, "Progress in scientific visualization," *The Visual Computer*, vol. 21, no. 12, pp. 979–984, 2005.
- [22] M. L. Connolly, *Molecular Surfaces: A Review*, Network Science, 1996.
- [23] M. L. Connolly, "Analytical molecular surface calculation," *Journal of Applied Crystallography*, vol. 6, pp. 548–558, 1983.
- [24] C. L. Bajaj, V. Pascucci, A. Shamir, R. J. Holt, and A. N. Netravali, "Dynamic maintenance and visualization of molecular surfaces," *Discrete Applied Mathematics*, vol. 127, no. 1, pp. 23–51, 2003.
- [25] P. Laug and H. Borouchaki, "Molecular surface modeling and meshing," *Engineering with Computers*, vol. 18, no. 3, pp. 199–210, 2002.
- [26] M. F. Sanner, A. J. Olson, and J.-C. Spehner, "Reduced surface: an efficient way to compute molecular surfaces," *Biopolymers*, vol. 38, no. 3, pp. 305–320, 1996.
- [27] H. Edelsbrunner and E. Mücke, "Three-dimensional alpha shapes," in *Proceedings of the Workshop on Volume Visualization*, pp. 75–82, ACM, Boston, Mass, USA, 1992.
- [28] D.-S. Kim, J. Seo, D. Kim, J. Ryu, and C.-H. Cho, "Three-dimensional beta shapes," *Computer-Aided Design*, vol. 38, no. 11, pp. 1179–1191, 2006.
- [29] J. Ryu, R. Park, and D.-S. Kim, "Molecular surfaces on proteins via beta shapes," *Computer-Aided Design*, vol. 39, no. 12, pp. 1042–1057, 2007.
- [30] H.-L. Cheng and X. Shi, "Quality mesh generation for molecular skin surfaces using restricted union of balls," *Computational Geometry*, vol. 42, no. 3, pp. 196–206, 2009.
- [31] Y. Zhang, G. Xu, and C. L. Bajaj, "Quality meshing of implicit solvation models of biomolecular structures," *Computer-Aided Geometric Design*, vol. 23, no. 6, pp. 510–530, 2006.
- [32] T. Can, C.-I. Chen, and Y.-F. Wang, "Efficient molecular surface generation using level-set methods," *Journal of Molecular Graphics and Modelling*, vol. 25, no. 4, pp. 442–454, 2006.
- [33] W. Lorensen and H. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, pp. 163–169, ACM, Anaheim, Calif, USA, July 1987.
- [34] E. Brigham and C. Yuen, "The fast Fourier transform," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, no. 2, p. 146, 1978.
- [35] H. Hoppe, T. DeRose, T. Duchamp, et al., "Piecewise smooth surface reconstruction," in *Proceedings of the 21st Annual*



*Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*, vol. 94, pp. 295–302, 1994.

- [36] D. Zorin, P. Schröder, and W. Sweldens, “Interpolating subdivision for meshes with arbitrary topology,” in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 189–192, ACM, New Orleans, La, USA, 1996.
- [37] E. F. Pettersen, T. D. Goddard, C. C. Huang, et al., “UCSF Chimera—a visualization system for exploratory research and analysis,” *Journal of Computational Chemistry*, vol. 25, no. 13, pp. 1605–1612, 2004.
- [38] N. Guex and M. Peitsch, “SWISS-MODEL and the Swiss-Pdb viewer: an environment for comparative protein modeling,” *Electrophoresis*, vol. 18, no. 15, 1997.