Computing Discrete Fine-Grained Representations of Protein Surfaces

Sebastian Daberdaku(✉) and Carlo Ferrari

Department of Information Engineering, University of Padova,
Via Gradenigo 6/B, 35131 Padova (PD), Italy
sebastian.daberdaku@dei.unipd.it, carlo.ferrari@unipd.it

Abstract. We present a voxel-based methodology for the computation of discrete representations of macromolecular surfaces at high resolutions. The procedure can calculate the three main molecular surfaces, namely van der Waals, Solvent-Accessible and Solvent-Excluded, by employing compact data structures and implementing a spatial slicing protocol. Fast Solvent-Excluded surface generation is achieved by adapting an approximate Euclidean Distance Transform algorithm. The algorithm exploits the geometrical relationship between the Solvent-Excluded and the Solvent-Accessible surfaces and limits the calculation of the distance map values to a small subset of the overall voxels representing the macromolecule. A parallelization scheme for the slicing procedure is also proposed and discussed.

Keywords: Macromolecular surface · High-resolution voxel surface · EDT

1 Introduction

Proteins are large molecules that play a vast range of biological functions by intervening in virtually all cellular activities. Each protein consists of one or more sequences of linear polymers (chains) of amino acids linked to each other by peptide covalent bonds. The physicochemical properties of its components, along with the surrounding environment (solvent), determine the proteins' specific three-dimensional (3D) shape. Proteins express their biological roles by binding selectively and with high affinity to other biomolecules, which depends on the formation of a set of weak, non-covalent bonds (hydrogen bonds, ionic bonds, van der Waals interactions) plus favourable hydrophobic interactions. Because the weakness of each individual bond, effective binding interactions require the simultaneous formation of multiple weak bonds, which is only possible if the surface contours of the interacting molecules are geometrically complementary. For this reason, many *in silico* methods for the prediction of protein functions, properties and interactions require proper representations of the molecular surface and its associated physicochemical properties.

To capture diverse aspects of the 3D geometry of proteins and macromolecules, different representations of the molecular surface have been introduced.

C. Angelini et al. (Eds.): CIBB 2015, LNBI 9874, pp. 180–195, 2016.
DOI: 10.1007/978-3-319-44332-4_14

Currently, the most used representations are: the van der Waals surface (vdW) [1], the Solvent-Accessible surface or Lee-Richards surface (SAS) [2] and the Solvent-Excluded surface (SES) or Connolly surface [3,4]. Surface calculations of large biomolecules, such as proteins or nucleic acids, based on their experimentally determined 3D structures (typically obtained by X-ray crystallography, NMR spectroscopy, or, increasingly, cryo-electron microscopy) have been extensively used in modern molecular biology studies [5–9].

Voxel-based molecular surface representations have received a lot of interest in many bioinformatics and computational biology applications. A voxel is the tiniest distinguishable element of a 3D object and can be thought as the equivalent of a pixel in the 3D domain. It is a discrete volume element that represents a single cell on a regularly spaced 3D grid. Voxels can be labelled with multiple values, describing various properties of a certain portion of space, and have been extensively used for visualization and analysis of scientific and medical data.

Voxelized protein surfaces are currently being employed in descriptor-based protein docking, pairwise alignment of molecules, protein shape comparison, pocket identification and FFT-based fast computations. Kihara et al. propose protein docking, shape comparison and interface identification methods based on 3D Zernike descriptors (3DZD) [10–12], which are calculated over circular surface patches of voxelized macromolecular surfaces. The voxelized representation of a molecular surface can describe the molecule's flexibility [13,14] and physicochemical property values, such as electrostatic potentials or hydrophobicity [15]. In [16] a ligand-binding pocket identification algorithm is introduced which uses a voxelized representation of the Connolly surface. In [17,18] the Fast Fourier Transform is used to efficiently match shape and electrostatic properties on surface grid points for protein docking. Protein surface atoms extraction based on a voxelized representation, which yields full atoms listings useful for studying binding regions on protein surfaces, was introduced in [19]. In [20], a voxelized protein representation is used for the identification and modelling of ligand binding areas. Grid representations of protein surfaces have also been used in cavity detection, binding-pockets identification and evaluation techniques [21–24].

Although macromolecular structural data repositories such as the Protein Data Bank (PDB) [25] have long been available, only a limited number of surface representations is provided, primarily aimed for visualization purposes. Surface calculation is an application-dependent task, resulting in multiple parametrisations based on the users' requirements. Protein surfaces are usually produced at runtime, adding high computational cost to the overall calculation. Moreover, many of the techniques and algorithms employing voxel-based molecular surfaces derive the latter from other explicit representations such as triangle mesh surfaces. Surface meshes are placed inside 3D grids, and the voxels whose centres are intersected by or within a certain distance from the mesh faces are marked as occupied (typically with $1, 0$ otherwise), resulting in tremendous accuracy loss for the final representation [26–28]. These naïve voxelization methods cannot guarantee two important requirements that voxelized surfaces must exhibit: separability and minimality [29]. The separability requirement ensures that the

resulting voxelized surface is connected and gap-free. On the other hand, a minimal voxelized surface should not contain voxels that, if removed, make no difference in terms of separability.

In this paper we propose and analyse a methodology for the computation of voxel-based fine-grained representations of the van der Waals, Solvent-Accessible and Solvent-Excluded surfaces starting directly from their PDB entries. To the extent of our knowledge, there are no available tools which can produce voxelized surface representations of macromolecules at the desired resolutions starting from their experimentally determined structural data (PDB entries). Several surface computation and visualization tools are available to date (see Table 1), but none of them provides voxelized representations of molecular surfaces. Representing and elaborating high-resolution 3D voxel grids requires memory-demanding data structures as well as high computational resources. We deal with the memory requirements by implementing a compact representation of the voxel grid and by implementing a spatial slicing protocol previously introduced in [30]. Only one bit is used to represent each voxel in the grid, which is eight times less than the smallest elemental type (char) on most systems. The molecule is sliced with parallel planes in a user-defined number of parts and the surface is computed for each slice separately. A parallelization scheme is also proposed and discussed. The parallel computation of voxelized surfaces on top of a compact data representation is the key to reducing computation time while maintaining accuracy, as shown by experimental results.

1.1 Macromolecular Surface Definitions

A molecule can be represented as a set of possibly overlapping spheres, each one having a radius equal to the van der Waals radius of the atom it represents. The topological boundary of this set of spheres is what is known as van der Waals surface. For proteins and other macromolecules, much of the van der Waals surface is buried in the inside of the molecules and is not accessible to the solvent or possible ligands. For this reason, the van der Waals surface is rarely used in bioinformatics applications. However, it is very important because it serves as a foundation to other surface definitions, and also because it is the basis of what is known as Corey-Pauling-Koltun (CPK) model (also known as calotte model or space-filling model [41,42]).

The Solvent-Accessible and Solvent-Excluded surfaces determine the 3D shape of the molecule in functional relationship with the external solvent. The Solvent-Accessible surface is defined as the geometric locus of the centre of a probe sphere (representing the solvent molecule) as it rolls over the van der Waals surface of the molecule. The Solvent-Excluded surface is defined as the locus of the inward-facing probe sphere as it rolls over the van der Waals surface of the molecule. This surface can be considered as a continuous sheet consisting of two parts: the contact surface and the re-entrant surface. The contact surface is part of the van der Waals surface that is accessible to a probe sphere. The re-entrant surface is the inward-facing surface of the probe when it touches two or more atoms. There is a clear relation between the SAS and the SES, as the

Table 1. Overview of some molecular surface computation tools.

Name	Surface representation			Comments
	vdW	SAS	SES	
PyMOL [31]	dot, spheres	dot, spheres	mesh	Scriptable molecular visualization system; extensible with Python
DeepView [32]	dot	dot	mesh	Tightly linked to SWISS-MODEL, an automated homology modelling server
MSMS [33]	n/a	n/a	dot, mesh	Dot surface over-sampled in some areas; can fail computing the surface of large molecules
UCSF chimera [34]	dot, spheres	dot, spheres	dot, mesh	Uses MSMS to compute the SES. Supports interactive visualization and analysis of molecular structures, density maps, assemblies, sequence alignments, docking results and trajectories
VMD [35]	dot, spheres	dot	mesh	Uses either SURF [36] or MSMS to compute the SES. Supports displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting
RasMol [37]	dot, spheres	spheres	n/a	Aimed at visualisation and generation of publication quality images
Jmol [38]	dot, spheres	dot, spheres	dot, mesh	Supports multiple molecules with independent movement, surfaces, orbitals, cavity visualization and crystal symmetry
Avogadro [39]	mesh	mesh	n/a	Advanced molecule editor; extensible via a plugin architecture
DS visualizer [40]	mesh	n/a	mesh	Commercial-grade graphics visualization tool for viewing, sharing, and analysing protein and modelling data

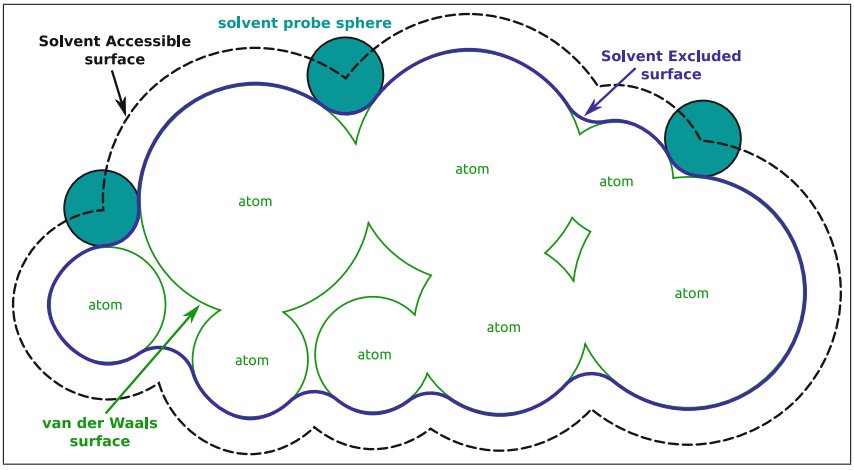

Fig. 1. The three main molecular surfaces: van der Waals (green), Solvent-Accessible (black) and Solvent-Excluded (blue). (Color figure online)

Solvent-Accessible surface is displaced outward from the Solvent-Excluded one by a distance equal to the probe radius. Figure 1 gives a graphical representation of the three molecular surfaces.

2 Materials and Methods

2.1 Surface Calculation Algorithm

The first step of the proposed methodology consists in acquiring the atomic coordinates of a macromolecule from its corresponding Protein Data Bank entry. The atomic coordinates of each atom composing the macromolecule and the relative radius are stored in a dedicated data structure. By default, the radius assignment for each atom type is based on the CHARMM27 force field [43], but users can provide their own radii information.

Pose normalisation follows the data acquisition step. The centre of gravity of the molecule is first translated to the coordinate origin. Then, a rotation is applied to the atomic coordinates in order to align the three principal axes of the molecule with the coordinate axes. The rotation matrix is determined by running Principal Component Analysis on the atomic coordinates.

The algorithm calculates the tightest axis-aligned bounding-box enclosing the whole molecule by determining the minimal and maximal atomic coordinates. Given a user-defined grid resolution parameter, the dimensions of the voxel grid which will contain the molecule are calculated. All atomic coordinates previously imported are translated, scaled and quantized to the new coordinate system defined by the voxel grid: each atom centre is mapped in its corresponding voxel in the voxel grid.

An adaptation of Bresenham's line algorithm [44] is used to efficiently determine the voxels occupied by a given atom. Each atom in the molecule is represented by a ball having a radius equal to either the atom's radius when calculating the vdW, or the atom's radius increased by the solvent-probe's radius when calculating the SAS and SES. After all the atoms composing the macromolecule have been mapped into the grid, we obtain the voxelized representation of the CPK model.

To obtain the van der Waals or the Solvent-Accessible surfaces, the boundary voxels of the voxelized representation of the CPK volumetric model of the macromolecule are extracted using an efficient 3D flood-filling algorithm [45]. The Solvent-Excluded surface is trickier to calculate because it includes the re-entrant surface portions. The proposed method is based on the Euclidean Distance Transform (EDT) algorithm for surface smoothing.

The implemented tool supports four different output formats: the Point Cloud Data file [46], OpenDX [47], Visualization Toolkit Structured Points and Visualization Toolkit PolyData [48,49].

2.2 The Euclidean Distance Transform

A distance transform (also known as distance map or distance field), is a derived representation of a digital image (usually a binary image). Distance maps are images where the value of each voxel of the foreground is the distance to the nearest voxel of the background. Let $B \in \{0,1\}^{l \times w \times h}$ be a binary voxel grid of length l, width w and height h. There are exactly $l \times w \times h$ voxels in B, each one identified by the ordered triple $\boldsymbol{v} = (i,j,k) \in V = \{1,\ldots,l\} \times \{1,\ldots,w\} \times \{1,\ldots,h\}$. Also, let $I_B : V \to \{0,1\}$ be the image function of B defined as

$$I_B(i,j,k) = b_{i,j,k} \in \{0,1\}\,, \tag{1}$$

where $b_{i,j,k}$ is the value of voxel $(i,\ j,\ k)$ in B. Let V_{O} be the set of occupied voxels of B, i.e.

$$V_{\mathrm{O}} = \{\boldsymbol{v} = (i,\ j,\ k) \in V \mid I_B(i,\ j,\ k) = 1\}\,. \tag{2}$$

Also, let $NBV_B : V \to V_{\mathrm{O}}$, such that $\forall \boldsymbol{v} \in V$, $NBV_B(\boldsymbol{v})$ is a nearest occupied voxel of B to $\boldsymbol{v}$, that is

$$NBV_B(\boldsymbol{v}) \in \underset{\boldsymbol{w} \in V_{\mathrm{O}}}{\arg\min}\, d(\boldsymbol{w},\boldsymbol{v}) = \{\boldsymbol{w} \in V_{\mathrm{O}} \mid \forall \boldsymbol{y} \in V_{\mathrm{O}} : d(\boldsymbol{w},\boldsymbol{v}) \leq d(\boldsymbol{y},\boldsymbol{v})\}\,, \tag{3}$$

according to some distance metric d. $NBV_B(\boldsymbol{v})$ is called the nearest boundary voxel (NBV) of $\boldsymbol{v}$ in B. Clearly, if $\boldsymbol{v} \in V_O$ then $NBV_B(\boldsymbol{v}) = \boldsymbol{v}$.

Finally, the distance transform of B (also known as distance map or distance field) is defined as a real-valued voxel grid $DT_B \in \mathbb{R}^{l \times w \times h}$ such that

$$I_{DT_B}(\boldsymbol{v}) = d(\boldsymbol{v}, NBV_B(v)),\ \forall\, \boldsymbol{v} \in V, \tag{4}$$

where $I_{DT_B} : V \to \mathbb{R}$ is the image function of DT_B.

When the chosen distance metric is the Euclidean Distance we talk about Euclidean Distance Transform (EDT). The Euclidean distance between two points $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^3$ is given by

$$d(\boldsymbol{v}, \boldsymbol{w}) = \|\boldsymbol{w} - \boldsymbol{v}\| = \sqrt{(w_x - v_x)^2 + (w_y - v_y)^2 + (w_z - v_z)^2}. \quad (5)$$

Squared Euclidean distance values among voxels are integers, and are often used to avoid time-consuming square root calculations.

2.3 Computing the Solvent-Excluded Surface

The Solvent-Excluded surface computation is based on the Euclidean Distance Transform. The employment of the Euclidean Distance Transform for macromolecular surface computation was first introduced in [50]. Let us consider the voxel grid containing the SAS and its relative Euclidean Distance Transform $EDT(SAS)$. Because the SAS is displaced outward from the SES by a distance equal to the probe radius, the latter can be obtained from the $EDT(SAS)$ by removing all voxels with a distance map value smaller than the probe radius from the CPK model, and then extracting the surface voxels of the resulting voxelized volume with the above-mentioned flood-filling algorithm.

To compute the SES, we only need distance values up to one probe-sphere radius from the SAS, and only for voxels inside the volume delimited by the SAS. We implemented the region-growing Euclidean Distance Transform algorithm described in [51] which can limit the computation up to a certain distance value and only to a given subset of the voxels in the grid. Starting from the voxels in the SAS, nearest boundary voxels are propagated towards the interior of the molecule. The boundary propagation is done considering voxels by increasing distance values, until the desired *depth* inside the Solvent-Accessible volume is reached.

The processing order of the voxels is enforced by a data structure called Hierarchical Queues (HQ), made of a collection of FIFO queues labelled from 0 to $d^2_{\max}$. Ingoing voxel in the HQ are inserted in the queue with label corresponding to their squared distance value. Outgoing voxels are extracted from the first non-empty queue with the least label. This way voxels are guaranteed to be parsed by increasing distance values. A map data structure is created to contain the squared distance values of each voxel. At the end of the procedure, this map will contain the squared Euclidean Distance Transform of the input voxel grid containing the SAS.

The HQ is initialised with queue 0 containing all the voxels belonging to the Solvent-Accessible surface, and all other queues empty. The map is initialised with 0 for all voxels belonging to the SAS and with MAXINT for all other voxels. Voxels are processed in the HQ-imposed order. For each voxel extracted from the HQ, its NBV is passed to its neighbours. If this leads to a smaller distance value than the previously stored one, the map is updated with the new value and the neighbour is inserted in the HQ. This procedure allows voxels to be mislabelled with a wrong NBV at first an then be corrected in a subsequent

step when a closer boundary voxel is found. The parsing order imposed by the HQ guarantees that errors are not propagated. Corrections are always processed before the initial errors propagate since they have smaller distance values and are placed in queues of smaller label.

Because the propagation is done only within a certain neighbourhood ($3 \times 3 \times 3$), certain voxels might be assigned an erroneous distance value. Erroneous distance values arise because the correct NBV are not propagated. Propagating boundary voxels to a larger neighbourhood ($5 \times 5 \times 5$) can significantly diminish the percentage of erroneous distance values, but also increases the time complexity of the algorithm. For this reason we adapted a two phase algorithm: the distance map is first computed quickly with the $3 \times 3 \times 3$ neighbourhood, and then the $5 \times 5 \times 5$ neighbourhood is used to correct errors made during the first scan. The corrections are required only for a small subset of voxels, i.e. the ones that were not propagated during the initial scan, and that have a distance value greater than a certain threshold.

2.4 The Slicing Procedure

To enable the computation of high resolution surfaces in spite of memory limitations we have developed a slicing protocol for the macromolecule. The molecule is sliced in a user-defined number of parts, and the surface is calculated separately for each part in a sequential fashion. The slicing is done with planes perpendicular to the x-axis of the Cartesian coordinate system (see Fig. 2).

Atom coordinates parsed from the PDB file are translated, scaled and quantized to the coordinate system defined by each slice. For each slice, we subtract the slice-length to the x coordinate of the translation vector $k-1$ times, where k is the current slice index ($k = 1, 2, ..., n$). The space filling procedure is performed for each slice separately, also taking into account any portions of atoms intersecting the slice whose centres might be located outside the current slice.

The correct determination of the distance map value for a given voxel requires knowledge of all boundary voxels within one probe sphere distance from the given voxel. Voxels in the immediate proximity of the slice borders require knowledge regarding the nearby boundary voxels in the adjacent slices in order to correctly calculate their distance map values. For this reason some extra margin on the x coordinate must be considered for each slice in order for the surface computation to yield correct results and it must be greater than the scaled and quantized probe-sphere radius (see Fig. 3).

Proteins can have solvent-accessible pockets which often serve as binding sites of other molecules. When running the spatial slicing procedure, pockets could run though two or more slices. Solvent-excluded voids buried inside the molecule can also be cut by the slicing planes, and there are cases when the surface portion belonging to a cavity cut by a slicing plane is disconnected from the outer molecular surface in that given slice. Communication among slices is required in order to correctly identify disconnected cavities cut by the slicing planes as solvent accessible or solvent excluded.

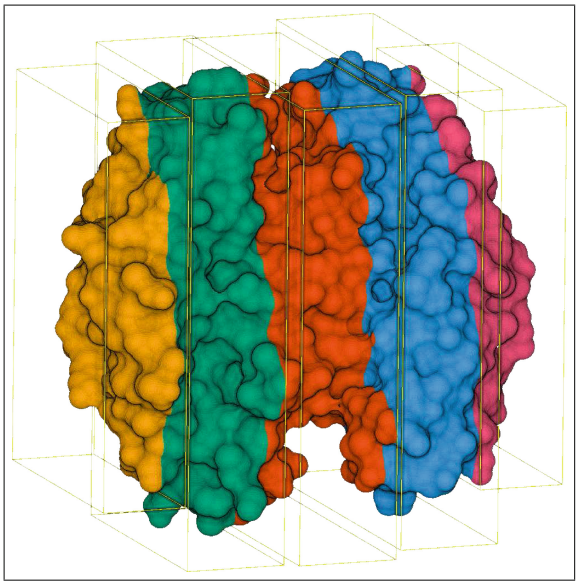

Fig. 2. Solvent-Excluded surface of 1VLA (4258 ATOM entries) [52] calculated with 5 slices, 1.4 Å probe-radius, 10^3 voxels per Å^3 resolution.

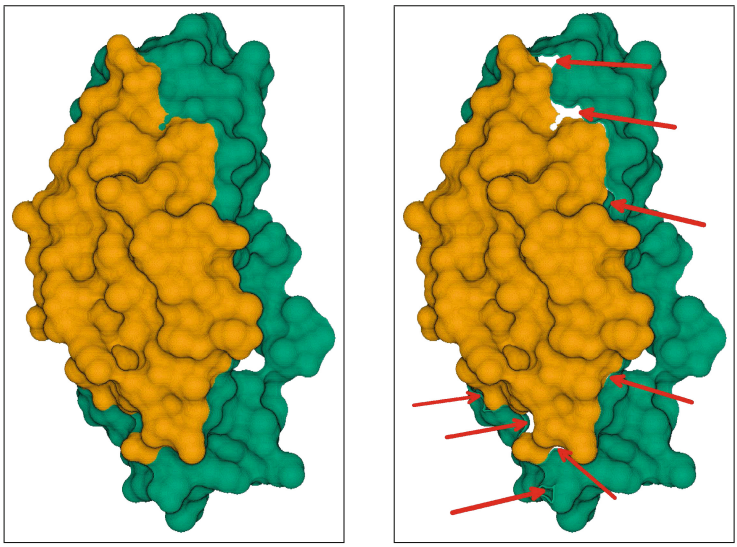

Fig. 3. Slices 1 and 2 of the SES of 1VLA, calculated with 5 slices, 1.4 Å probe-radius, 10^3 voxels per Å^3 resolution. We can see the differences between surfaces computed with (left) and without (right) the slice margin.

The procedure can be described as follows. Candidate pocket cavities are identified by checking in the margin region of each slice for free solvent-excluded voxels. The algorithm extracts all surface voxels of potential pockets from each slice using an adaptation of the same efficient 3D flood-filling procedure mentioned earlier. Pockets and solvent-excluded cavities running through two or more slices must be distinguished from each other. The algorithm extracts all surface voxels belonging to potential pockets from each slice and stores them in an apposite data structure. For each pair of adjacent slices, the border voxels of the candidate pockets are matched against their neighbours on the other slice. A candidate pocket is solvent-accessible if its border voxels have free neighbours on the adjacent slice. Otherwise, if its border voxels are matched with the border voxels of another candidate pocket on the adjacent slice, the current pocket remains undetermined as it could run through two or more slices in length. Multiple border exchange iterations are required if there are large pockets or cavities that run through more than one slice in length in the current macromolecule. The procedure ends when, at a given iteration, no new candidate pockets are recognized as solvent accessible.

2.5 Parallelization

The macromolecular surface calculation protocol with slicing introduced in the previous sections suggests an immediate parallelization scheme. The surface calculation for each slice can be executed nearly-independently from the others, as process synchronization and communication is required only for the pocket-detection and extraction procedure, in order to correctly identify pockets spanning between two or more slices.

3 Results and Discussion

We have run different tests of an MPI-based implementation of the parallel algorithm on an IBM®Power®P770 Server with 6 IBM®Power7 3.1 GHz CPUs and 640 Gb of RAM, running SUSE Linux Enterprise 11, and experimentally determined the average computation times for different input molecules at various resolutions, while calculating the three molecular surfaces. For the tests we used molecules 1GZX (4387 ATOM records) [53] and 2AEB (9568 ATOM records) [54], see Fig. 4.

3.1 Workload Distribution

To obtain an equitable workload distribution among processes, the slicing procedure should guarantee a uniform distribution of the workload. A uniform distribution of the number of atoms per slice (i.e. variable-length slices) yields better speedup values than employing a constant slice length value.

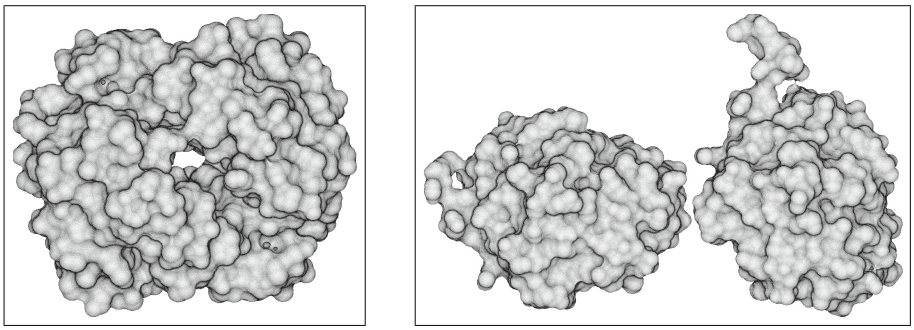

Fig. 4. SES of 1GZX (left) and 2AEB (right) computed with 1.4 Å probe-radius and 10^3 voxels per Å^3 resolution.

3.2 Surface Computation Time

The average surface computation times that follow were calculated over 100 runs of the corresponding configuration (PDB entry, desired molecular surface, resolution, probe radius, number of processes). We progressively increased the number of processes (from 1 to 64) and evaluated the mean computation time for each configuration (see Fig. 5). The tests were conducted for very high resolution molecular surface representations.

The overall speedup is affected by the constant margin introduced in each slice during the SES computation. At some point, while increasing the number of processors (increasing number of slices), the overhead introduced by the margins will become comparable to the slice computation time, thus vanishing the benefits of further parallelization.

3.3 Sequential Slicing

To overcome memory limitations the method implements a compact representation of the voxel grid and uses a sequential spatial slicing procedure: the molecule is sliced by a user-defined number of parallel planes perpendicular to the x-axis and the surface is computed for each slice sequentially. For instance, the calculation of the surface for 1GZX at a resolution of 9000 voxels per Å^3 and while dividing the molecule in 10 slices, needs nearly 5 GB of RAM, against the 2.6 GB used while dividing the molecule in 20 slices (tests were made on a desktop computer with an Intel Core i7 860 CPU and 8 GB of RAM (4×2GB DDR3-1333 banks) running Ubuntu 13.10 $\times$ 64), which is an easily affordable amount of memory nowadays in desktop computers. By tuning the resolution and number-of-slices parameters, various memory utilization rates can be achieved depending on the users' needs.

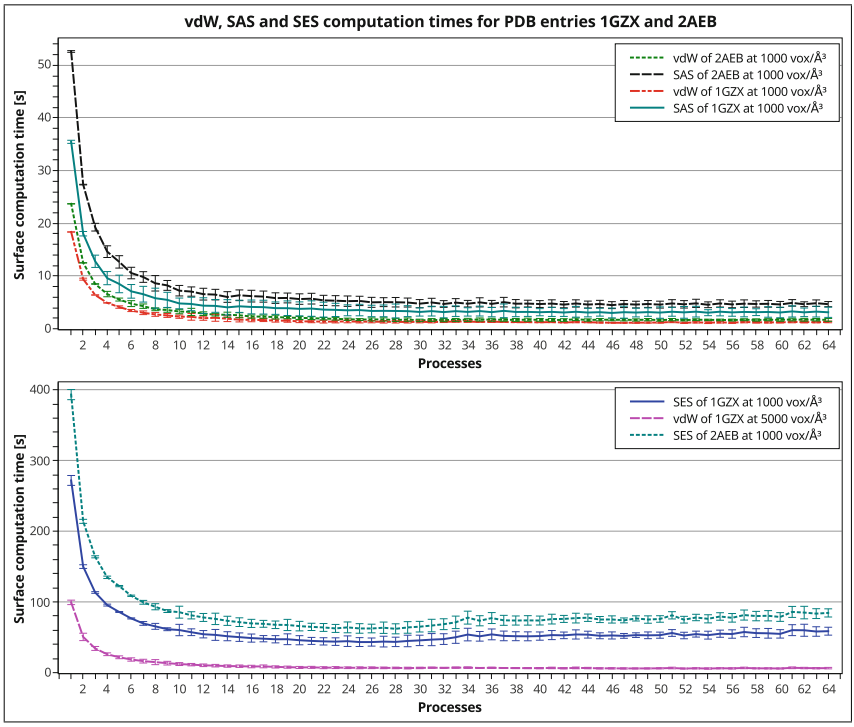

Fig. 5. Surface computation time for 1GZX and 2AEB for various configurations. Error bars represent one standard deviation of the computation time based on 100 repetitions.

3.4 Separability and Minimality

The voxelized representations of molecular surfaces produced by this approach possess both separability and minimality properties. The voxelized surface is extracted using a flood-filling algorithm: any gap would cause leakage of the flood through the discrete surface. Also, if any voxel in the resulting surface is removed, the flood-filling algorithm would propagate through that voxel towards the outside of the space-filling volume. This means that the produced surface is both separating and minimal.

4 Conclusion

We presented a methodology for the computation of discrete fine-grained voxelized macromolecular surfaces, implemented in a tool that can generate the three main molecular surfaces at high-resolutions effectively and in a timely manner. This makes it perfectly suitable for integration into other applications or pipelines of bioinformatics tools which require the computation of such surfaces at runtime. The parallel implementation introduces advantages in terms

of the overall speedup, however the uniform distribution of atoms per slice may not necessarily yield a balanced workload between processes. On the other hand, the constant slice margin represents the main limitation to this parallelization scheme as it introduces constant overhead regardless of the slice size. These issues are left for future work.

Voxelized representations are well suited to represent multiple physicochemical and geometrical properties of molecular surfaces, as each voxel can describe multiple properties of a portion of the 3D space. Although only binary voxels were employed in this work, they could easily be extended to contain multiple values in order to represent several properties, such as electrostatic potentials, hydrophobicity, curvature, surface normals, etc. For instance, we are currently developing a local surface descriptor for protein-protein docking based on the voxelized representation of shape and electrostatic properties of the molecular surface. The descriptor is invariant to roto-translations and allows the efficient comparison of geometric and electrostatic complementarity between surface portions.

Linux binaries and test results are available at: http://www.dei.unipd.it/~daberdak/VoxSurf

Acknowledgements. This work has been partially supported by the University of Padova ex60 % grant "Advanced Applications in Computer Science".

The authors are grateful to Prof. Gianfranco Bilardi and Dr. Paolo Emilio Mazzon for their help in using the facilities of the Advanced Computing Paradigms Lab.

References

1. Whitley, D.C.: Van der Waals surface graphs and molecular shape. J. Math. Chem. **23**(3–4), 377–397 (1998)
2. Lee, B., Richards, F.: The interpretation of protein structures: estimation of static accessibility. J. Mol. Biol. **55**(3), 379–IN4 (1971)
3. Connolly, M.L.: Analytical molecular surface calculation. J. Appl. Crystallogr. **16**(5), 548–558 (1983)
4. Connolly, M.L.: The molecular surface package. J. Mol. Graph. **11**(2), 139–141 (1993)
5. Sanner, M.F., Olson, A.J., Spehner, J.C.: Fast and robust computation of molecular surfaces. In: Proceedings of the Eleventh Annual Symposium on Computational Geometry, SCG 1995, pp. 406–407. ACM, New York (1995)
6. Mitchell, J.C., Kerr, R., Ten Eyck, L.F.: Rapid atomic density methods for molecular shape characterization. J. Mol. Graph. Model. **19**(3–4), 325–330 (2001)
7. Kinoshita, K., Nakamura, H.: Identification of the ligand binding sites on the molecular surface of proteins. Protein Sci. **14**(3), 711–718 (2005)
8. Bock, M.E., Cortelazzo, G.M., Ferrari, C., Guerra, C.: Identifying similar surface patches on proteins using a spin-image surface representation. In: Apostolico, A., Crochemore, M., Park, K. (eds.) CPM 2005. LNCS, vol. 3537, pp. 417–428. Springer, Heidelberg (2005)
9. Albou, L.P., Schwarz, B., Poch, O., Wurtz, J.M., Moras, D.: Defining and characterizing protein surface using alpha shapes. Proteins: Struct. Funct. Bioinf. **76**(1), 1–12 (2009)

10. Venkatraman, V., Yang, Y., Sael, L., Kihara, D.: Protein-protein docking using region-based 3D Zernike descriptors. BMC Bioinform. **10**(1), 407 (2009)
11. Li, B., Kihara, D.: Protein docking prediction using predicted protein-protein interface. BMC Bioinform. **13**(1), 7 (2012)
12. Esquivel-Rodriguez, J., Filos-Gonzalez, V., Li, B., Kihara, D.: Pairwise and multimeric protein-protein docking using the LZerD program suite. Protein Struct. Predict. **1137**, 209–234 (2014)
13. Grandison, S., Roberts, C., Morris, R.J.: The application of 3D Zernike moments for the description of "Model-Free" molecular structure, functional motion, and structural reliability. J. Comput. Biol. **16**(3), 487–500 (2009)
14. Kihara, D., Sael, L., Chikhi, R., Esquivel-Rodriguez, J.: Molecular surface representation using 3D Zernike descriptors for protein shape comparison and docking. Curr. Protein Pept. Sci. **12**(6), 520–533 (2011)
15. Sael, L., La, D., Li, B., Rustamov, R., Kihara, D.: Rapid comparison of properties on protein surface. Proteins **73**(1), 1–10 (2008)
16. Huang, B., Schroeder, M.: Ligsite^{csc}: predicting ligand binding sites using the connolly surface and degree of conservation. BMC Struct. Biol. **6**(1), 1–11 (2006)
17. Kozakov, D., Brenke, R., Comeau, S.R., Vajda, S.: PIPER: an FFT-based protein docking program with pairwise potentials. Proteins: Struct., Funct., Bioinf. **65**(2), 392–406 (2006)
18. Chowdhury, R., Rasheed, M., Keidel, D., Moussalem, M., Olson, A., Sanner, M., Bajaj, C.: Protein-protein docking with F^{2}Dock 2.0 and GB-rerank. PLoS ONE **8**(3), e51307 (2013)
19. Lee, L.W., Bargiela, A.: Protein surface atoms extraction: voxels as an investigative tool. Eng. Lett. **20**(3), 217–228 (2012)
20. Lee, L.W., Bargiela, A.: An approximated voxel approach for the identification and modelling of ligand-binding sites. J. Phys. Sci. Appl. **2**(10), 399–408 (2012)
21. Levitt, D.G., Banaszak, L.J.: POCKET: a computer graphics method for identifying and displaying protein cavities and their surrounding amino acids. J. Mol. Graph. **10**(4), 229–234 (1992)
22. Hendlich, M., Rippmann, F., Barnickel, G.: LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J. Mol. Graph. Model. **15**(6), 359–363 (1997)
23. Weisel, M., Proschak, E., Schneider, G.: PocketPicker: analysis of ligand binding-sites with shape descriptors. Chem. Cent. J. **1**(1), 7 (2007)
24. Li, B., Turuvekere, S., Agrawal, M., La, D., Ramani, K., Kihara, D.: Characterization of local geometry of protein surfaces with the visibility criterion. Proteins: Struct. Funct. Bioinform. **71**(2), 670–683 (2008)
25. Berman, H., Henrick, K., Nakamura, H.: Announcing the worldwide protein data bank. Nat. Struct. Mol. Biol. **10**(12), 980 (2003)
26. Sael, L., Li, B., La, D., Fang, Y., Ramani, K., Rustamov, R., Kihara, D.: Fast protein tertiary structure retrieval based on global surface shape similarity. Proteins: Struct., Funct., Bioinf. **72**(4), 1259–1273 (2008)
27. La, D., Esquivel-Rodrguez, J., Venkatraman, V., Li, B., Sael, L., Ueng, S., Ahrendt, S., Kihara, D.: 3D-SURFER: software for high-throughput protein surface comparison and analysis. Bioinformatics **25**(21), 2843–2844 (2009)
28. Axenopoulos, A., Daras, P., Papadopoulos, G., Houstis, E.: A shape descriptor for fast complementarity matching in molecular docking. IEEE/ACM Trans. Comput. Biol. Bioinf. **8**(6), 1441–1457 (2011)

29. Huang, J., Yagel, R., Filippov, V., Kurzion, Y.: An accurate method for voxelizing polygon meshes. In: 1998 IEEE Symposium on Volume Visualization, pp. 119–126. IEEE (1998)
30. Daberdaku, S., Ferrari, C.: A voxel-based tool for protein surface representation. In: Angelini, C., Bongcam-Rudloff, E., Decarli, A., Rancoita, P.M., Rovetta, S., (eds.) Computational Intelligence Methods for Bioinformatics and Biostatistics, CIBB 2015, Naples, Italy, pp. 96–101 (2015)
31. Schrödinger, L.L.C.: The PyMOL molecular graphics system, version 1.8, November 2015
32. Guex, N., Peitsch, M.C.: SWISS-MODEL and the Swiss-Pdb viewer: an environment for comparative protein modeling. Electrophoresis **18**(15), 2714–2723 (1997)
33. Sanner, M.F., Olson, A.J., Spehner, J.C.: Reduced surface: an efficient way to compute molecular surfaces. Biopolymers **38**(3), 305–320 (1996)
34. Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., Ferrin, T.E.: UCSF chimera–a visualization system for exploratory research and analysis. J. Comput. Chem. **25**(13), 1605–1612 (2004)
35. Humphrey, W., Dalke, A., Schulten, K.: VMD – visual molecular dynamics. J. Mol. Graph. **14**, 33–38 (1996)
36. Varshney, A., Brooks Jr., F.P., Wright, W.V.: Computing smooth molecular surfaces. IEEE Comput. Graphics Appl. **14**(5), 19–25 (1994)
37. Sayle, R.A., Milner-White, E.J.: RASMOL: biomolecular graphics for all. Trends Biochem. Sci. **20**(9), 374–376 (1995)
38. Jmol: an open-source Java viewer for chemical structures in 3D. http://www.jmol.org/
39. Hanwell, M.D., Curtis, D.E., Lonie, D.C., Vandermeersch, T., Zurek, E., Hutchison, G.R.: Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. J. Cheminformatics **4**(1), 17 (2012)
40. BIOVIA, Dassault Systèmes: Discovery Studio Modeling Environment, Release 4.5 (2015)
41. Corey, R.B., Pauling, L.: Molecular models of amino acids, peptides, and proteins. Rev. Sci. Instrum. **24**(8), 621–627 (1953)
42. Koltun, W.L.: Precision space-filling atomic models. Biopolymers **3**(6), 665–679 (1965)
43. MacKerell, A.D., Bashford, D., Bellott, M., Dunbrack, R.L., Evanseck, J.D., Field, M.J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kuczera, K., Lau, F.T.K., Mattos, C., Michnick, S., Ngo, T., Nguyen, D.T., Prodhom, B., Reiher, W.E., Roux, B., Schlenkrich, M., Smith, J.C., Stote, R., Straub, J., Watanabe, M., Wirkiewicz-Kuczera, J., Yin, D., Karplus, M.: All-atom empirical potential for molecular modeling and dynamics studies of proteins. J. Phys. Chem. B **102**(18), 3586–3616 (1998)
44. Bresenham, J.: Algorithm for computer control of a digital plotter. IBM Syst. J. **4**(1), 25–30 (1965)
45. Yu, W.W., He, F., Xi, P.: A rapid 3D seed-filling algorithm based on scan slice. Comput. Graph. **34**(4), 449–459 (2010). Procedural Methods in Computer Graphics Illustrative Visualization
46. Rusu, R.B.: The pcd (point cloud data) file format. http://pointclouds.org/documentation/tutorials/pcd_file_format.php. Accessed 27 Apr 2016
47. Thompson, D., Braun, J., Ford, R.: OpenDX: Paths to Visualization; Materials Used for Learning OpenDX the Open Source Derivative of IBM's Visualization Data Explorer. Visualization and Imagery Solutions, Missoula (2004)

48. Schroeder, W., Martin, K., Lorensen, B.: Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th edn. Kitware, New York (2006)
49. Avila, L., Kitware, I.: The VTK User's Guide. Kitware, New York (2010)
50. Xu, D., Zhang, Y.: Generating triangulated macromolecular surfaces by Euclidean distance transform. PLoS ONE **4**(12), e8140 (2009)
51. Cuisenaire, O.: Region growing Euclidean distance transforms. In: Bimbo, A. (ed.) ICIAP 1997. LNCS, vol. 1310, pp. 263–270. Springer, Heidelberg (1997)
52. Joint Center for Structural Genomics (JCSG): Crystal structure of Hydroperoxide resistance protein OsmC (TM0919) from Thermotoga maritima at 1.80 Å resolution (2004)
53. Paoli, M., Liddington, R., Tame, J., Wilkinson, A., Dodson, G.: Crystal structure of T state haemoglobin with oxygen bound at all four haems. J. Mol. Biol. **256**(4), 775–792 (1996)
54. Di Costanzo, L., Sabio, G., Mora, A., Rodriguez, P.C., Ochoa, A.C., Centeno, F., Christianson, D.W.: Crystal structure of human arginase I at 1.29 Å resolution and exploration of inhibition in the immune response. Proc. Nat. Acad. Sci. U.S.A. **102**(37), 13058–13063 (2005)