VAPLI: Novel Visual Abstraction for Protein-Lipid Interactions

Naif Alharbi
Swansea University, UK *

Michael Krone
University of Tübingen, Germany †

Matthieu Chavent
IPBS, Toulouse, France ‡

Robert S. Laramee
Swansea University, UK §

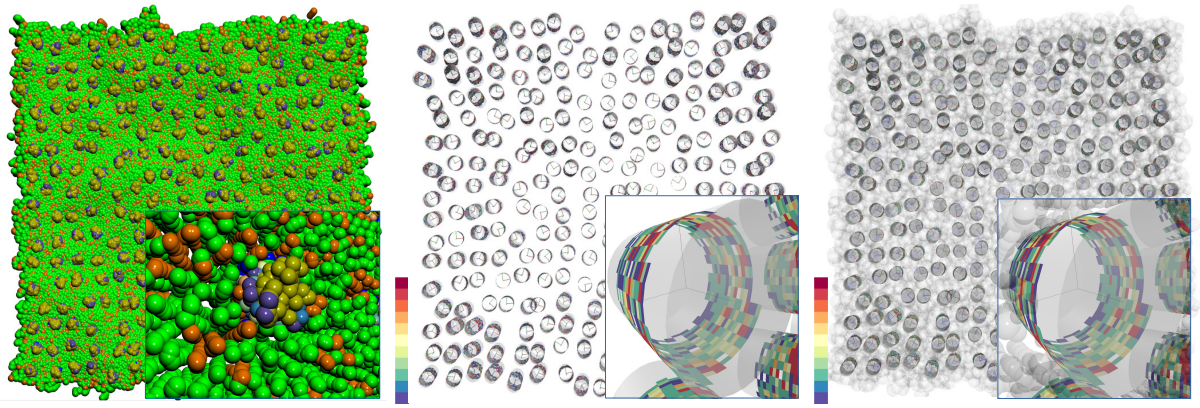

Figure 1: Visual Abstraction for Protein-Lipid Interaction (VAPLI). (Left) A naive visualization of protein-lipid interaction (PLI). (right) Focus-and-Context applied to the PLI. Our PLI depiction is rendered with lipid particles, using transparent Van der Waals representation, to illustrate the membrane context. (middle) The PLI space is depicted using only our VAPLI approach, focusing on proteins greatly reducing both complexity and occlusion. Color is mapped to the frequency of PLI on a tiled cylinder.

Abstract

Molecular visualization of molecular dynamics data commonly uses representative surfaces of varying resolution to depict protein molecules while a variety of geometric shapes, from lines to spheres, are often used to represent atoms and inter-atomic bonds. In general, the aim of molecular visualization is focused on efficiently rendering atoms or molecules themselves, while the interaction space between them is less explored. Furthermore, with naive approaches rendering every molecule, the particles overlap so that significant interaction details are obscured due to clutter. Contrary to common approaches, our work focuses on the interaction between lipids and proteins and the area in which this occurs, the - lipid - membrane. To do so, we introduce a novel abstract interaction space for Protein-Lipid Interaction (PLI). The cylindrical abstraction simplifies perception and computation of PLI. It does so by using a local projection of PLI onto a cylindrical geometry. This also addresses the challenge of visualizing complex, time-dependent interactions between these molecules. We propose a fast GPU-based implementation that maps lipid-constituents involved in the interaction onto the abstract protein interaction space. The tool provides interactive rendering of PLI for 336,260 particles over almost 2,000 time-steps. The result is a great simplification of this complex Protein-Lipid Interaction leading to both better perception and new insight for computational biologists.

Index Terms: Visualization—Molecular Dynamics—Protein-Lipid Interaction

*668696@swansea.ac.uk
†michael.krone@uni-tuebingen.de
‡Matthieu.Chavent@ipbs.fr
§r.s.laramee@swansea.ac.uk

1 Introduction and Motivation

Visualization of Molecular Dynamics (MD) simulations has received a great deal of attention over the past decades. Numerous representations are available for the depiction of MD data from simple molecules to complex macromolecular systems [1, 13]. Protein molecules, for example, are often represented by surfaces [5, 14, 20] while small molecules can be depicted as licorice, ball-and-stick or Van der Waals spheres [4]. These representations generally take every particle constituting the molecules into account which gives a very accurate picture of the molecular systems. However, in the case of a very big and crowded environment, such as numerous proteins embedded in a large patch of lipid membrane, these same representations might also become inadequate due to computational complexity or by occluding useful information presented to the user. Here, we would like to propose a new type of abstraction in order to gain insight into very large model systems constituted by numerous protein and lipid molecules.

Membrane proteins perform a wide range of biological functions such as drug transporters, ion channels, or receptors transmitting molecules and information across the cell membrane. These proteins are often dynamically modulated by their environment, i.e., the surrounding membrane. This modulation can be mediated through global effects such as the membrane curvature [2] or rely on more subtle changes like close interactions in between lipids and proteins [27]. MD simulation is now a powerful tool to assess such interactions [9]. However, it can be difficult to fully understand the molecular mechanisms of such interactions especially for large systems. Furthermore, these interactions are often transitory and quickly change in a time-dependent manner. For example, the dynamics and complexity of membrane models cause many particles to be occluded or to overlap (see left image in Figure 1). This is creating a number of computation and visualization challenges which may not be overcome by classical molecular depictions.

Our solution is to design a new type of molecular abstraction focusing on the Protein-Lipid Interactions (PLI). This representation enables the depiction of numerous membrane proteins while retaining useful information held by the PLI. We design and implement

this Visual Abstraction for PLI (VAPLI) in close collaboration with domain experts to help computational biologists obtain insight into their membrane time-dependent systems. Our contributions are:
1) A novel cylindrical definition and abstraction of the PLI for large MD simulations,
2) A fast GPU-based implementation to calculate PLI on the fly,
3)The analysis and feedback of an expert in computational biology to our novel PLI representation.
In short, the frequency of PLIs is represented by a tiled cylinder. Color-mapped tiles indicate PLI frequency. The paper is organized as follows: in Section 2, we discuss works related to molecular visualization and abstraction. In Section 3, we describe the data from MD simulations we are using. Section 4 discusses the design and implementation of our VAPLI approach. The preliminary results are presented in Section 5 following by the domain expert feedback in Section 6. The conclusions and future work are provided in Section 7.

2 Related Work

There are a number of recent and helpful survey papers on molecular visualization. O'Donoghue *et al.* [24] review visualization methods and tools that enable the community of life scientists to gain insight into their molecular structure data. In a recent state-of-the-art report, Kozlíková *et al.* [13] review and classify visualization techniques developed to render molecular structures. Alharbi *et al.* [1] present a survey of surveys on molecular dynamics visualization data focusing on challenges for computational biologists which may to be solved by computer graphics approaches.

In the field of molecular visualization using abstractions is mainly used to: 1) enhance software performance, or 2) reduce visual complexity, and enhance perception. Here, we classify the abstraction research into two groups based on the proposed approach: abstractions used to increase the computing performance and abstractions to improve user perception.

Abstractions To Accelerate Performance And Reduce Computational Complexity: In this subsection, we discuss previous work that incorporates abstraction and level-of-detail (LoD) techniques to represent large data-sets relying on spatial configuration approaches to accelerate performance. Previous literature often studies spatial configuration approaches that rely on 1) the arrangement of the atoms of a molecule, or 2) manipulating the resolution of the surface of a molecule with respect to the LoD. A common goal is to enable a user to visualize large molecular datasets interactively by reducing the representation of the data based on the desired level of detail.

TexMol [3] introduces a biochemically sensitive level-of-detail hierarchy for molecular representations. A single sphere is used to represent an atom with the atom's Van der Waal radius, whereas a residue is represented by a minimal single bounding sphere that encloses all of its component atoms. This approach reduces the visual clutter by communicating the appropriate volume occupancy and shape. In addition, their hierarchical image-based rendering enables mapping of derived physical properties onto molecular surfaces.

Sharma *et al.* [26] implement a hierarchical view frustum-culling algorithm based on an octree data structure. The algorithm efficiently removes atoms outside of the user's field-of-view.

Lee *et al.* [18] propose a view-dependent LoD technique for real-time surface rendering. It provides high-quality rendering and is able to display critical parts of molecular models. Lampe *et al.* [16] introduce a two-level approach to visualize large, dynamic protein complexes. In the first level, each residue is replaced with a single vertex based on the residue's rigid transformation. In the second level, the atoms which are contained in the residues are dynamically generated in the geometry shader based on the initial simulation vector and the transformation matrix.

Lindow *et al.* [19] introduce a GPU-based, 3D voxel grid to store the atomic data and utilize a fast ray-voxel traversal by which only spheres in the current voxel are tested for intersection. Similar to Lindow *et al.* [19], Falk *et al.* [7] store each molecule in its own supporting grid, which is then traversed with a level-of-detail via a ray. The grid is constructed based on the size of each atom whereas in [19] the grid is based on the number of atoms. Krone *et al.* [15] propose a novel fast molecule surface extraction approach which can be used to depict structural detail on a continuous scale. By manipulating the grid resolution and the density kernel function the scale can range from the atomic detail to reduced detail visual representations.

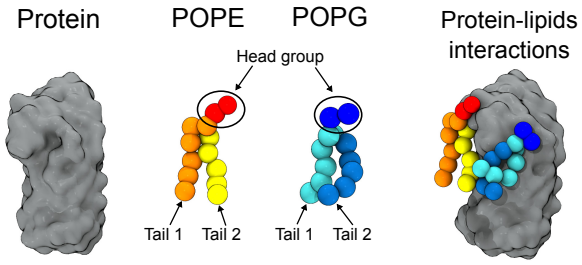

Figure 2: Typical representations of molecules: protein (surface) on left, POPG and POPE lipids (Van der Waals) in the middle, and two protein-lipids interactions (on right). Protein and lipids representations were generated with VMD [10].

Le Muzic *et al.* [17] present a LoD approach that summarizes adjacent atoms into a single sphere depending on the distance to the camera. They utilize a texture to store the atom positions of a whole molecule. Their approach is similar to Lampe *et al.* [16] but uses the tessellation shader instead of the geometry shader.

Abstractions To Enhance User Perception: This subsection discusses work providing different visual abstractions of a molecule with respect to its structure to enhanced user perception. In these cases, each representation is generally associated with a specific molecular structure and different LoD techniques are applied to provide a smooth transition between structures.

Weber [29] proposes a texture-based approach using shader programming to generate pen-and-ink renderings of molecules in real-time. Van der Zwan *et al.* [28] propose a novel continuous abstraction space for illustrative molecular visualization. The method is GPU-based for visualizing continuous transitions between different stages of structural abstraction of a molecular system (i.e., space-filling, ball-and-stick, licorice, backbone, and ribbon). Parulek *et al.* [25] propose a novel LoD approach to visualize large protein complexes. The application of individual levels is based on the distance to the camera. They employ three different surface representations (solvent-excluded surface (SES), Gaussian kernels and Van der Waals spheres). Mohammed *et al.* [23] present a novel tool for visualizing astrocytes and neurons at different levels of abstraction. The tool benefits from a novel abstraction space to provide a set of visualizations ranging from a detailed 3D surface renderings of segmented structure (realistic images) to simplified abstractions (e.g. skeletons and graphs).

In this work, we introduce a novel abstraction of the protein-lipid interaction space to address the visual complexity and perception of such a representation and provide a simplified depiction of the PLI. This representation supports detecting PLIs on-the-fly by using GPU-accelerated calculations.

3 Simulation Data Description

The dataset we have used represents a trajectory of a large patch of a membrane constituted by lipids and proteins extracted from molecular dynamics a simulation. The simulation is used to study the dynamic behaviour of both proteins and lipids in the context of a membrane aligned primarily in the *xy* plane [6]. The dimensions of

the system are $116 \times 116 \times 10$ nanometers (x, y, and z respectively) and represent individual trajectories of 336,260 particles over almost 2 microseconds (stored in c.a. 2000 frames). The system consists of three molecule types: one protein, and two lipid types (POPE and POPG) (see Figure 2). There are 256 protein molecules while the number of POPE and the POPG lipid molecules are 14,354 and 4,738 respectively. The models presented are based on the MARTINI coarse-grained forcefield [22] which does not represent all the atoms but simplifies 4 heavy atoms by one coarse-grained (CG) particle. Thus the hierarchy for each type of molecule is simplified:
Each protein is constituted by 171 residues and each residue is composed of 1 to 3 particles. The total number of particles per protein is 344 resulting in 88,064 particles in total (256×344).
The total number of lipids (POPG and POPE) is 19,092. Each lipid molecule has 13 particles which result in ($19{,}092 \times 13$) 248,196 particles in total. For this representation, we divide each lipid into 3 groups: a head group of 2 particles, a first tail group of 6 particles, and a second tail group of 5 particles (see Figure 2).

In terms of the data size, the simulation contains more than 666 million space-time positions that occupy 8 Gigabytes of memory.

4 Visualization of Protein-Lipid Interaction (PLI)

In this section, we present why and how we have designed the PLI space as a cylinder surrounding the protein. We present different aspects of the implementation of the PLI abstraction including PLI testing, translation and rotation of the cylinder, and grid tiling.

Design: The interaction space is represented by a tiled cylinder. The main axis of the cylinder, $\mathbf{e}_z$, is aligned with the principal axis of the molecule, it surrounds, and is updated at each time step. The orientation captures the main component of lipid translation that influences PLI. The PLI frequency is reflected in the local tailing. The tiles are color-mapped based on how often a lipid crosses into the approximate molecular space. This is not typical as there are no previous abstractions that are dedicated to interaction space that we are aware of.

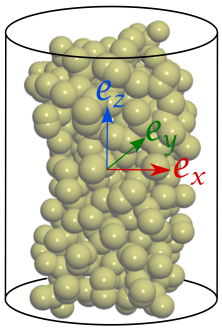

Figure 3: Principal components of a protein interaction space at $t = 0$. The SVD method is used to calculate the *eigenvectors* ($\mathbf{e}_x, \mathbf{e}_y$, and $\mathbf{e}_z$) to initialize the local cylinder and its extent from the original protein particles.

Defining the PLI Representation: We chose a cylinder surrounding each protein to represent the PLI space. This simplified abstraction is a good approximation of membrane protein shape (see Figures 2 and 3) and features at least two advantages in comparison to a complex molecular surface: i) simplifying the visualization and ii) enhancing the computing performance.

To create the cylinder, we first calculate the Center of Mass (CoM) and the eigenvalues for each protein (see Equations 1, 2, 3). We then position the center of the cylinder onto the protein CoM. By default, the largest eigenvalue is used as the cylinder height while the second eigenvalue defines the cylinder radius (Figure 3). The cylinder height and radius can be also specified by the user to interactively manipulate the PLI space size. The cylinder mesh is triangulated on the GPU based on the protein CoM, and values of height and radius. The opacity of the mesh can also be adapted interactively to provide different levels of transparency.

We provide two representations, a static representation and a dynamic representation. The key difference between the two representations is that the static representation omits the x and y-axis rotation.

Updating the PLI Representation: The dynamics of protein particles play a significant role in defining the geometry of a protein at a given time step. Some protein properties (e.g. global position, CoM, principal axes, spatial extent, and rotation) might be influenced by the system dynamics. As a result, the PLI space can be updated for every time step. We use Singular-Value Decomposition (SVD) [12] to calculate the main properties of a protein interaction space i.e., the principal axes (*eigenvectors*), dimension sizes (*eigenvalues*) and rotation matrix. SVD requires the covariance matrix C of a protein space. C is a symmetric real $n \times n$ matrix. There is an orthogonal matrix V and diagonal Σ such that:

$$C = V \Sigma V^T \tag{1}$$

The diagonal entries of Σ, that is $\Sigma_{ii} = \sigma_i$, can be arranged to be non-negative in order of decreasing magnitude. Positive entries are called the singular values (*eigenvalues*) of C. The columns of V and V^T are called left and right singular vectors (*eigenvectors*) and form an orthonormal basis for $\boldsymbol{R^n}$ [11]. The eigenvectors represent the axes of a protein interaction space and the associated eigenvalues are used to define a protein interaction space extent. The first eigenvector is used to represent the z axis orthogonal to the primary protein bi-layer, and the second and third eigenvector are used to represent the x axis and y axis - the orientation of the dominant lipid motion. Figure 3 shows the eigenvectors of a protein interaction space. However, the order of the second and the third eigenvector is not necessarily constant over time and needs to be consistent. This challenge is addressed by performing two tests, the first test is performed at $t = 0$ to ensure that the x axis and the y axis are in the correct order. The second test is performed at every time step $t > 0$ to maintain consistency of the x and y axes.

PLI Cylinder Translation and Rotation: In the model of membrane studied here, each protein molecule is constituted by a finite number of particles (344 particles). These particles are linked together through constraints defined by the forcefield which limits the overall displacement of one particle towards the others belonging to the same protein. Thus, the overall protein movement can be simplified by the movement of its Center of Mass (CoM). For each time step, the CoM of a protein is calculated using the following:

$$CoM(t) = \frac{\sum_{i=1}^{n} p_i(t)}{n} \tag{2}$$

Where $p(t)$ is the position of particle p at a given time step t and n is the number of particles. The trajectory of the CoM of a protein represents the evolution of the protein position over the simulation time. The rotation of a protein is defined by the internal motion of its particles. Two successive positions of all the protein's particles are needed to calculate the covariance matrix C of a protein.

$$C(t) = \sum_{i=1}^{n} p_i(t) \times p_i^T(t+1) \tag{3}$$

$p(t)$ is the position of particle p at a given time step t and p^T is the transpose vector of the position of particle p at $t = t+1$. Applying Equation 1 to C we get V and V^T. The rotation matrix R can be calculated from V and V^T:

$$R(t) = V(t)V^T(t) \tag{4}$$

The translation and the rotation are then applied to the cylinder mesh to update the cylinder position.

PLI Calculation: To calculate PLI, we provide two tests (see Figure 4) in order to simplify the overall calculations. First, we approximate

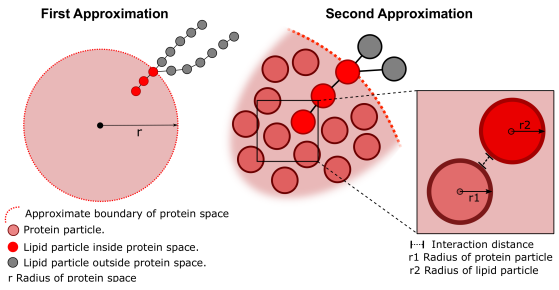

Figure 4: The left image depicts our first approximation for the PLI test. The test requires only the distance between a lipid particle and the principal axis of a protein. The right image illustrates the second approximation for PLI: for every lipid particle that passes the first test, another distance test between the given lipid particle and the protein particles can be performed.

the protein shape by the cylindrical region used to depict the PLI space defined previously. If the smallest distance between one lipid particle and the central axis of the protein is less than the cylinder radius we consider that the particle is interacting with the protein (first approximation). If this first test is accepted then the distance between every individual protein particle and the given lipid particle can be calculated for higher accuracy (second approximation). In this case, we consider an interaction between protein and lipid particles if the distance between the two is below a given threshold distance in the range of $0.6-0.8$ nm. The minimum and maximum values are defined in Marrink et al [21] and Goose et al [8] respectively. So we took the largest value to be sure that we have at least one particle interacting with the protein residues.

Tiling the Interaction: The interaction test result, for each protein, is stored in a uniform 2D grid structure. The grid size is 32×32 by default or can be specified by the user. The x-axis of the 2D grid represents the angle of PLI cylinder while the y-axis represents its height (see Figure 5). While the number of grid cells can vary, the grid has a fixed length along the x-axis of 2π and a fixed height h defined previously. Every cell of the 2D grid defines a protein area where the PLI is calculated and updated. In this paper, we use a regular 2D grid rather than a quad-tree structure. Nevertheless, we would like to implement such a structure in future work.

Projecting the Interaction onto the Abstract Protein Space: The interaction mapping process is performed on the GPU by using the geometry shader. The advantage of this approach is that it is no longer necessary to store, update, translate, or rotate the interaction positions for every time-step because they are based on the given local protein coordinate system. For the full detail representation, all the non-null cells, i.e cells that contain interaction detail, are projected as a color-mapped tile onto the surface of the respective protein space. Each tile represents the frequency with which a lipid has intersected its extent. The bottom-left corner of the projected patch is defined by the bottom, left edge of the cell. The size of the projected patch is based on the tiling. See Figure 5.

5 Preliminary Results

The complexity of PLI increases over time which requires special visualization rather than naive approaches. Figure 1 shows the visualization of complex simulation data and compares our approach to a naive approach. The left image depicts a crowded environment of lipids and proteins with a naive approach. The middle image visualizes the data based on our abstraction. Focus-and-context is used in the right image. One can easily see that in the naive visualization most of the PLI details are obscured due to lipid occlusion. Our approach provides an overview of the PLIs for the whole system and the user can easily zoom in to obtain more details about a particular PLI. With our approach, some patterns emerge: PLI can form a wide-band of interactions on the abstracted surface. The highest

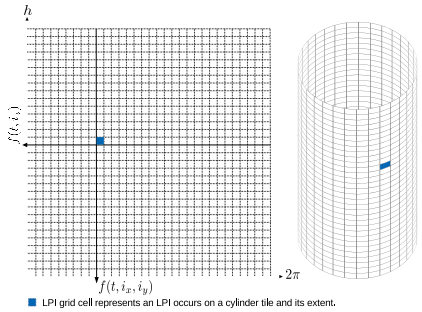

Figure 5: PLI interaction frequency is stored in a uniform grid. The highlighted cell represents a PLI tile at the highest resolution. Tiles are used to depict PLI at different resolutions. The x-axis represents the PLI angular value at a given time step t. The y-axis represents the z component of PLI at t.

frequency of PLI occurs at different locations but are mainly focused in the center of the cylinder while the upper and bottom edges are less populated reflecting the strong interactions between the tails of the lipids and the proteins. A supplementary video showing the results presented in this paper can be found at:
`https://youtu.be/Nnb7yzZN2HY`

6 Domain Expert Feedback

We have been working closely with domain experts in molecular dynamics since 2015. All of the designs presented here are inspired by our discussions with them. Here is direct feedback from one of them. "*This new PLI abstraction allows the user to quickly understand where the lipids can interact with each protein. The main interactions are visible in the hydrophobic parts in the center of the proteins which is expected but was not seen so clearly with more classical representations. In general, it was necessary to perform a costly post-treatment on a part of the trajectory or on the whole trajectory to get the same information. Furthermore, using this PLI abstraction gave a very quick idea of where to focus the attention on interesting proteins which can strongly interact with the lipids.*"

7 Conclusions and Future Work

Numerous tools have been designed to better understand structures of molecules from simple molecules to complex macromolecular systems. Few tools are focusing on time-dependent datasets from Molecular Dynamics simulations and none are focusing on visualizing the molecular interactions between lipids and protein. In this work, we introduce a novel abstract PLI space to address computational and visualization challenges arising from lipid-protein dynamics. This representation highlights PLI better than classical naive representations and allows experts to quickly highlight interesting molecular features. Thus, this representation provides great potential to investigate PLI from molecular dynamics systems. We are now planning to assess our method on more complicated test cases with 5 to 10 types of lipids interacting with different proteins. We also plan to design a quad-tree structure to handle the interaction data. We also would like to use LoD techniques on lipid molecules to reduce overlap and occlusion. We will also add new features to help the user to easily compare different features of two or more proteins such as using a 2D projection.

Acknowledgments

This work was partially funded by the Ministry of Education of Saudi Arabia, the Saudi Cultural Bureau in London, the Department of Computer Science at Swansea University, and the DFG as part of project PROLINT. Finally, we would like to thank D. Rees and L. McNabb for proof-reading the paper.

REFERENCES

[1] N. Alharbi, M. Alharbi, X. Martinez, M. Krone, A. Rose, M. Baaden, R. S. Laramee, and M. Chavent. Molecular visualization of computational biology data: A survey of surveys. *Eurovis short papers*, pp. 133–137, 2017.

[2] B. Antonny. Mechanisms of Membrane Curvature Sensing. *Annual review of biochemistry*, 80(1):101–123, July 2011.

[3] C. Bajaj, P. Djeu, V. Siddavanahalli, and A. Thane. Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. In *Proceedings of the Conference on Visualization'04*, pp. 243–250. IEEE Computer Society, 2004.

[4] M. Chavent, B. Lévy, M. Krone, K. Bidmon, J.-P. Nominé, T. Ertl, and M. Baaden. Gpu-powered tools boost molecular visualization. *Briefings in Bioinformatics*, p. bbq089, 2011.

[5] M. Chavent, B. Lévy, and B. Maigret. MetaMol: high-quality visualization of molecular skin surface. *Journal of molecular graphics & modelling*, 27(2):209–216, Sept. 2008.

[6] M. Chavent, T. Reddy, J. Goose, A. C. E. Dahl, J. E. Stone, B. Jobard, and M. S. Sansom. Methodologies for the analysis of instantaneous lipid diffusion in md simulations of large membrane systems. *Faraday discussions*, 169:455–475, 2014.

[7] M. Falk, M. Krone, and T. Ertl. Atomistic visualization of mesoscopic whole-cell simulations using ray-casted instancing. *Computer Graphics Forum*, 32(8):195–206, 2013.

[8] J. E. Goose and M. S. Sansom. Reduced lateral mobility of lipids and proteins in crowded membranes. *PLoS Comput Biol*, 9(4):e1003033, 2013.

[9] G. Hedger and M. S. Sansom. Lipid interaction sites on channels, transporters and receptors: recent insights from molecular dynamics simulations. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 2016.

[10] W. Humphrey, A. Dalke, and K. Schulten. VMD: visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–38, 1996.

[11] D. Kalman. A singularly valuable decomposition: the svd of a matrix. *The College Mathematics Journal*, 27(1):2–23, 1996.

[12] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.

[13] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum*, 36(8):178–204, 2017.

[14] M. Krone, K. Bidmon, and T. Ertl. Interactive visualization of molecular surface dynamics. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1391–1398, 2009.

[15] M. Krone, J. E. Stone, T. Ertl, and K. Schulten. Fast visualization of gaussian density surfaces for molecular dynamics and particle system trajectories. *EuroVis Short Papers*, 2012:67–71, 2012.

[16] O. D. Lampe, I. Viola, N. Reuter, and H. Hauser. Two-level approach to efficient visualization of protein dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1616–1623, 2007.

[17] M. Le Muzic, J. Parulek, A.-K. Stavrum, and I. Viola. Illustrative visualization of molecular reactions using omniscient intelligence and passive agents. In *Computer Graphics Forum*, vol. 33, pp. 141–150, 2014.

[18] J. Lee, S. Park, and J.-I. Kim. View-dependent rendering of large-scale molecular models using level of detail. In *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, vol. 1, pp. 691–698. IEEE, 2006.

[19] N. Lindow, D. Baum, and H.-C. Hege. Interactive rendering of materials and biological structures on atomic and nanoscopic scale. In *Computer Graphics Forum*, vol. 31, pp. 1325–1334, 2012.

[20] N. Lindow, D. Baum, S. Prohaska, and H.-C. Hege. Accelerated visualization of dynamic molecular surfaces. *Computer Graphics Forum*, 29(3):943–952, 2010.

[21] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. De Vries. The martini force field: coarse grained model for biomolecular simulations. *The journal of physical chemistry B*, 111(27):7812–7824, 2007.

[22] S. J. Marrink and D. P. Tieleman. Perspective on the martini model. *Chemical Society Reviews*, 42(16):6801–6822, 2013.

[23] H. Mohammed, A. K. Al-Awami, J. Beyer, C. Cali, P. Magistretti, H. Pfister, and M. Hadwiger. Abstractocyte: A visual tool for exploring nanoscale astroglial cells. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):853–861, 2018.

[24] S. I. O'Donoghue, D. S. Goodsell, A. S. Frangakis, F. Jossinet, R. A. Laskowski, M. Nilges, H. R. Saibil, A. Schafferhans, R. C. Wade, E. Westhof, et al. Visualization of macromolecular structures. *Nature Methods*, 7:S42–S55, 2010.

[25] J. Parulek, D. Jönsson, T. Ropinski, S. Bruckner, A. Ynnerman, and I. Viola. Continuous levels-of-detail and visual abstraction for seamless molecular visualization. In *Computer Graphics Forum*, vol. 33, pp. 276–287. Wiley Online Library, 2014.

[26] A. Sharma, R. K. Kalia, A. Nakano, and P. Vashishta. Scalable and portable visualization of large atomistic datasets. *Computer Physics Communications*, 163(1):53–64, 2004.

[27] H. Sprong, P. van der Sluijs, and G. van Meer. How proteins move lipids and lipids move proteins. *Nature reviews. Molecular cell biology*, 2(7):504–513, 2001.

[28] M. Van Der Zwan, W. Lueks, H. Bekker, and T. Isenberg. Illustrative molecular visualization with continuous abstraction. In *Computer Graphics Forum*, vol. 30, pp. 683–690, 2011.

[29] J. R. Weber. Proteinshader: illustrative rendering of macromolecules. *BMC Structural Biology*, 9(1):19, 2009.