EG UK Computer Graphics & Visual Computing (2016)
Cagatay Turkay and Tao R. Wan (Editors)

MolPathFinder: Interactive Multi-Dimensional Path Filtering of Molecular Dynamics Simulation Data

Naif Alharbi[1] , Robert S. Laramee[1] , Matthieu Chavent[2]

[1]Swansea University, Department of Computer Science, [2]University of Oxford, Department of Biochemistry

Abstract

Molecular Dynamics Simulations (MDS) play an important role in the field of computational biology. The simulations produce large high-dimensional, spatio-temporal data describing the motion of atoms and molecules. A central challenge in the field is the extraction and visualization of useful behavioral patterns from these simulations. Many visualization tools have been proposed to help computational biologists gain insight into MDS data. While recent developments focused on accelerating and optimising the rendering, it is still necessary to design new metaphors to better understand and filter MDS datasets. In this article, we are describing a set of tools to interactively filter and highlight dynamic and complex paths constituted by motions of molecules. In collaboration with computational biologists, we have tested our approach on large-scale, real data. Based on the user's feedback, our program helped scientists to navigate more easily through their dataset and isolate interesting patterns. Furthermore, our approach was useful to investigate both local and global behavior of molecular motions.

Categories and Subject Descriptors (according to ACM CCS): Scientific visualization [Human-centered computing]: Applied computing—Molecular evolution

1. Introduction

Nowadays, advances in simulations allow the design of large models constituted by million, or billion of atoms Perilla et al [PGC*15]. Furthermore, modelling very large membrane systems is becoming a subject of intense research. With current methods, one can perform membrane organelles, virus envelopes, and large patches of bacterial membranes Chavent et al [CDS16]. These new models are indeed especially difficult to analyze due to the diversity of molecules - particularly the lipids - and the huge quantity of data. Thus, computational biologists are now facing challenges to find programs that can help them to visualize and analyze their MDS data. Molecular simulations have benefitted from the development of a range of visualization tools: from the well established VMD [HDS96] or Pymol [DeL02] viewers to the more recently developed Megamol [GKM*15] or Unitymol [LTDS*13] programs. These programs take advantage of advances in hardware technologies such as GPUs capabilities Chavent et al [CLK*11]. Nevertheless, there are still tremendous efforts needed to convey clear information about MDS trajectories from atomic details to a more global scale. In this context, the analysis of lipids dynamical behavior in membrane models is a typical example. At the atomic level, it is important to understand how lipids and proteins can finely interact together and how it can influence their respective dynamical properties Hedger and Sansom [HS16]. Zooming out, at the level of the whole model, aggregates of lipids move together as swarms. At this scale, the dynamical behavior of these aggregates need to be taken into account Chavent et al [CRG*14]. Our goal is to gain insights into the complex motions of lipids molecules at both scales. Common molecular viewers do no have yet dedicated rendering functions to analyze such lipid paths in a user-friendly way Chavent et al [CRG*14]. Thus, we have developed MolPathFinder as a set of tools to help the user interactively extract paths features during the course of the Molecular Dynamics simulation. This program exploits new interactive visualization techniques:

- Focus and context visualization techniques that exploit the GPU to help guide the users attention to specific regions of interest.
- Novel interactive filtering techniques to help the user to identify trajectories based on path length, edge length, curvature and normalized curvature, and their combinations.
- Combination of 2D-3D path rendering in a dual dimension representation in order to highlight differences arising from the 2D projection on a plane.

We applied our approach on a real test case used to better understand the formation of protein clusters at the surface of bacteria Rassam et al [RCB*15]. In this work, the lipid dynamics was not extensively studied due to the lack of available tools. In this article, we will show how one can use MolPathFinder to quickly gain insights about the MDS data.
The paper is organized as follows: In section 2, we give an overview of tools dedicated to the analysis of MDS data at different scales. In section 3, we present the available features of MolPathFinder and

DOI: 10.2312/cgvc.20161289

how they were designed. In section 4, we briefly describe our first results and the user feedbacks.

2. Related Work

In biology, visualizing data is a key step in order to gain insight into mechanisms and functions of cells and organelles. In numerous scientific fields related to biology, it is now possible to extract a vast amount of data. This creates new challenges for scientists to manage and decipher this deluge of information and requires the development of new tools O'Donoghue et al [OGG*10]. Structural biology generates more and more complex and large data: from protein structure to nucleic acid assemblies. Thus, structural biologists are using a wide range of programs to make sense of their data O'Donoghue et al [OGF*10]. These programs are often dedicated to render static molecules via a wide range of metaphors like balls-and- sticks, ribbons, or even simple lines Goodsell [Goo05]. From these static structures it is then possible to create dynamic models using the so-called Molecular Dynamics methodology Karplus and Petsko [KP*90]. The visualization of trajectories from molecular dynamics simulations has received significant attention over the last few decades Kozlíková et al [KKL*15]. To the list of well established and widely used programs - such as PyMol [DeL02], VMD [HDS96], Chimera [PGH*04],YASARA [Kri03], [KV14] - we can add recent developments like MegaMol [GKM*15] or UnityMol [LTDS*13]. All these programs have numerous functionalities to analyze and display the MDS data. They are also optimized to harness recent hardware facilities to efficiently render time dependent datasets and not only static structures (Chavent et al. [CLK*11], Hirst et al. [HGB14]).
Nevertheless, these programs rarely propose new ways of visualizing the data. They rely on known metaphors such as Surface, Van der Waals, or secondary-structures representations. With the increase of larger models obtained by MD simulations (Chavent et al. [CDS16], Perilla et al. [PGC*15]), it is necessary to define new representations in order to 1) simplify and clarify the visualization, and 2) limit the number of graphical primitives to accelerate the rendering. With these constraints in mind, we designed MolPathFinder as a set of tools to display the path of molecules and map dynamical properties onto it. This type of rendering is widely used in Physical sciences Lipşa et al. [LLC*12] but is not yet adopted in computational biology. Recent works has begun to use this approach to display small molecules movements at the surface of proteins Bidmon et al. [BGB*08] and nucleic acids Ertl et al. [EKK*14] or in the cavities Lindow et al [LBH11]. Here, we are using path lines as a descriptor to render the large number of lipid molecules and how they can dynamically move together.

Our multi-dimensional tools focus on MDS trajectories by quantifying their attributes such as length, edge length, curvature and normalized curvature. The tools provide the user with a number of different interactive filtering techniques. In addition to the 2D and 3D overview of the molecular dynamics, color mapping is used to enable the user to visualize the trajectories and to identify their distribution. Different filtering techniques are used to render a subset of the trajectories based on their length, edge length, curvature and normalized curvature. We provide the user with focus and context techniques to de-emphasize less interesting data. Finally, all the features of the trajectories are computed in both 2D and 3D. A novel visualization method is used to compare the properties of the trajectories in 2D and 3D.

3. Visualization of MDS Data with MolPathFinder

User Requirements Nowadays, there is a large range of available tools to create and analyze membrane models Javanainen and Martinez-Seara [JMS16]. Therefore, it is now possible to quickly create very large membrane models. Nevertheless, there are rooms to develop new programs to better visualize and interact with such models. One of the main difficulties is to interactively visualize and explore the data. To do so, it is essential to remove unnecessary details while keeping a meaningful rendering. Thanks to our collaboration with computational biologists, we defined the paths of molecules as a good descriptor to tackle this issue. Commonly used programs, such as VMD, can efficiently display very large systems but, to our knowledge, have no build-in function to easily manipulate and decipher molecular paths (see Figure 1). MolPathFinder has been developed and designed to fulfil this requirement. A particular attention was paid to offer a variety of features to be mapped onto the path: from the length to the curvature of the path. This visualization helps the user observe the distribution of paths based on a chosen attribute and provides the user with a clear overview of the local properties of the trajectories. However, color mapping alone does not always scale well with data size. We also used filtering techniques to help the user highlight areas of interests and only focus on relevant features. The filtering techniques will be described in section 3.1.2 but first we are describing basic functions required to load data and select different parts of the molecule

Loading the data Molecular dynamics simulations produce different formats. Each format is designed and proposed for a specific purpose. For example, the GROMACS [VDSLH*05] software

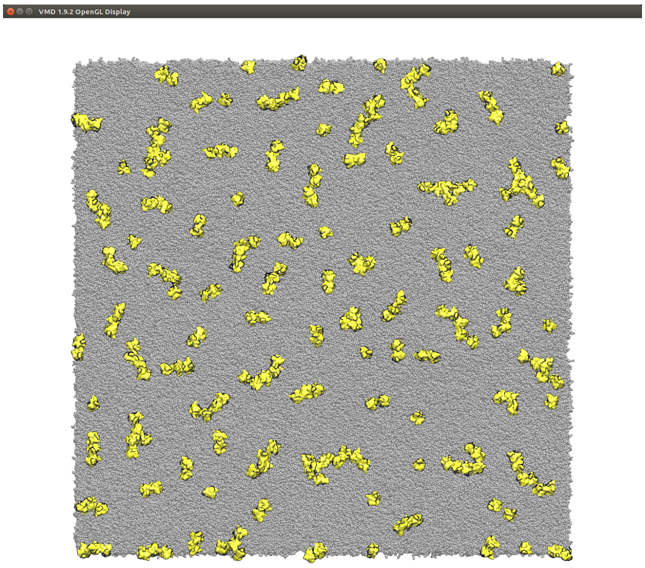

Figure 1: Snapshot of the VMD program. The protein data is represented by yellow surfaces and the lipids by gray spheres.

provides coordinate, trajectory, energy and topology data. Our data contains the trajectories of 242,790 atoms over a time span of 51 nanosecond (ns).

GRO File The associated GRO file is utilized to get the atoms details i.e. residue name and atom name and to construct the data hierarchy. Only lipid molecules were present in the GRO file provided by our collaborator. There were 2 types of lipids in this file called POPG and POPE. Each lipid molecule is then constituted by several atoms, for example the POPE residue includes 13 atom types C1A, C1B, C2A, C2B, C3A, C4A, C4B, C5B, D3B, GL1, GL2, NH3 and PO4. The hierarchy is used to build the data exploration window (see Molecules explorer and Figure 2).

XTC File In order to read the trajectory file, we utilize the GROMACS xdrLibrary which is designed to enable developers to read and write .xtc, .edr and .trr files [GRO09]. The computer simulations define a set of boundary conditions to approximate a large system by using a set of small cells (each cell is called a unit cell). If an atom crosses one side of the unit cell, it enters the opposite side with the same velocity. We process the periodic boundary conditions (PBCs) to introduce spatial domain boundaries and the original coordinate system of the atoms is converted into euclidean space coordinates.

Molecules Explorer We design an interface to facilitate the process of atom-type selection. See Figure 2. The data hierarchy is used to present a GUI control that enables the user to select one or more atoms from different residues. Each atom can be colored by a user-chosen color map. One of the advantages of this tool is that it helps the user comparing the behaviors of two or more atoms from different amino acids.

3.1. Visualization Techniques and Data Representation

In this subsection we introduce three visualization techniques and briefly discusses the methods that we use to derive the atom trajectory attributes. The atom and path representations, and the color mapping are first described. Then, the focus and context technique in Section 3.1.1. The filtering techniques are described in the sections 3.1.2 to 3.1.7, and the dual space representation in section 3.1.8.

Atom and Path Representations The user is able to represent the trajectories in three ways. 1) depicting atoms only, 2) depicting paths only and 3) depicting atoms and their paths simultaneously. We provide the user with both static and dynamic representations of the trajectories. Atoms are represented by spheres, and curves are used to represent the atom path. See Figure 3.

Color Mapping We utilize color mapping to map local length, edge length, curvature and normalized curvature to a variety of color scales. The nature of our data can exploit a diverging color scheme to easily distinquish between the different values. In addition to the sequential color scheme and qualitative color schemes, ColorBrewer [Col09] provides nine different diverging color schemes. The user can change two coloring options, 1) the scheme control and 2) the class number control. The former is to switch between the color schemes and the latter is to select the number of colors. The color scheme is interactively mapped and applied to the trajectories based on a user-chosen property.

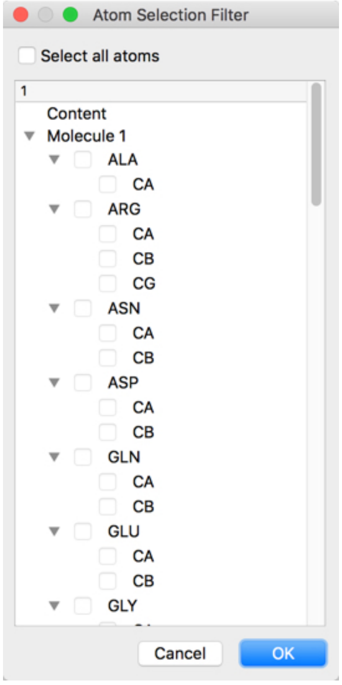

Figure 2: Atom selection filter: the molecule hierarchy is utilized to construct the control. The user is able to select different atoms from different residues (amino acids).

3.1.1. Focus + Context

Our collaborators provided a very large dataset. Sometimes, the user wants to direct his attention only on a small area of interest. To let him select this type of area we used a Focus + Context technique. We utilize this technique in such a way that it combines with the interactive filtering techniques. The focus data is rendered in its original color whereas a transparent gray is used to render the context. See Figure 4.

3.1.2. Path Filtering Techniques

Atom trajectories have a number of characteristic properties such as edge length, total length and curvature. Each property conveys different information and requires specific computation. Color mapping is used to represent the local magnitude of these properties. The filtering techniques examine the paths to de-emphasize trajectories that do not match the user selection.

Even though the Focus + Context technique is a very useful approach, it may require the user to select a region of space. However, the different path attributes such as edge length, total length and curvature require special filtering. The user must be able to specify these properties to reduce the rendered data. Based on one or more of the chosen attributes, the path filtering techniques discard paths that do not fulfil the user criteria. The user is provided with range slider controls to specify minimum and maximum ranges. The trajectories are filtered interactively. The filtering techniques work in

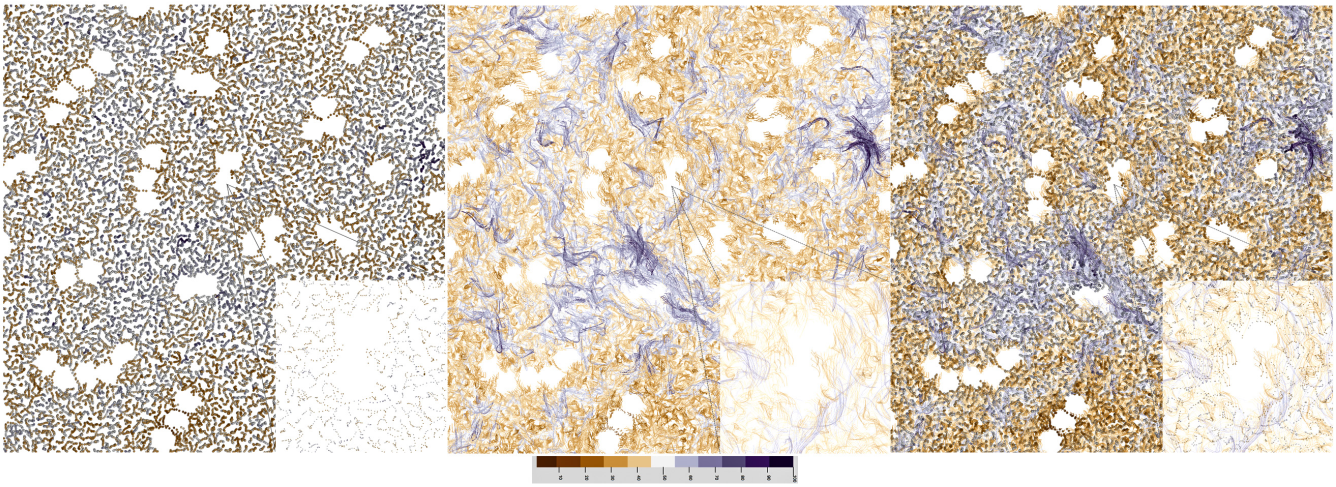

Figure 3: Atoms and their path representations. Atoms represented as spheres at time step 0 (left), the atoms' path represented by lines (middle), and the representation of atoms' path and the atoms at time step 51 (right). Color is mapped to total path curvature in 3D.

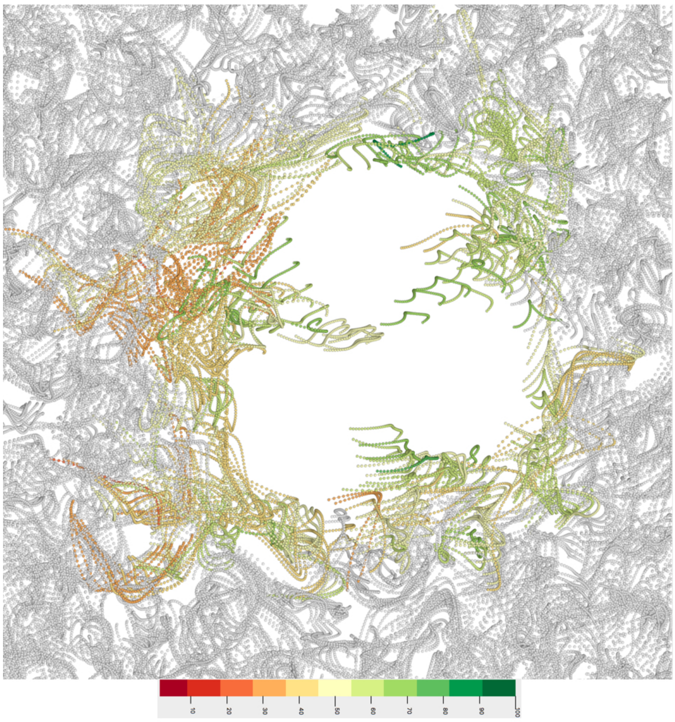

Figure 4: A combination of focus + context and color mapping. The focus shows paths surrounding a protein. The total path curvature is coded by color in 3D. The context is de-emphasized by a transparent gray color.

conjunction with both the color mapping and the focus + context visualization.

3.1.3. Filtering Based on Local Edge Length of Paths

In time-dependent trajectory data, the length of path edges reflects the atoms velocity throughout the trajectory. Understanding how the molecules are moving together can be very important to better decipher their aggregation. It is especially true for lipid molecules Apajalahti et al [ANG*10]. Filtering the paths in function of their length will clearly highlight groups of molecules with equivalent dynamical properties hence allowing the user to visually delimit groups of molecules. Our trajectory data involves 51 time steps for each atom. Each pair of points in a trajectory constructs an edge. For 2D edges and 3D edges, the edge length is simply computed using the distance between successive points, p_i, p_{i+1}, for each point $p \in \mathbb{R}^3$ in the x,y,z spatial domain. Let T be a trajectory and let p_i and p_{i+1} be two sequential points on trajectory T. The edge $\overline{e}$ is constructed from p_i and p_{i+1}. See Figure 5.

$$\overline{e} = p_{i+1} - p_i$$

We find the magnitude $|\overline{e}|$ of edge $\overline{e}$ using:

$$|\overline{e}| = \sqrt{(p_{i+1})^2 - (p_i)^2}$$

We store the result of this computation for the local path length. The result for each atom path is normalized by the maximum edge length $|\overline{e}|_{max}$ and stored in the GPU's vertex buffer for color mapping. The current published approaches for visualizing the edge of an atom path fall into two catagoriez 1) a glyph-based approache (using arrows) and 2) a connected edges approach. Chavent et al. [CRG*14] combine the two representations into a single depiction. They represent a path using cylinders. The cylinders' position is mapped to time.

We propose a similar approach to Chavent et al. [CRG*14] by depicting the path as connected edges combined with varying atom size. This approach helps in both the overview representation and lateral displacement representation. See Figure 6.

3.1.4. Filtering Based on Total Path Length

As the local edge length filtering, the total path filtering will help the user to quickly define groups of molecules with the

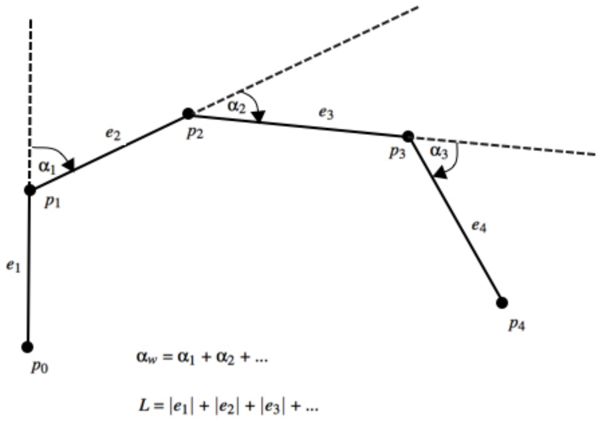

Figure 5: The winding-angle α_w is the sum of the angles between the edges e_i and e_{i+1}. The path length L is the sum of the magnitudes $|\overline{e}|$ of the path's edges. Image based on Sadarjoen and Post [SP00].

same dynamical features. The edge length of all paths is already computed (see the previous subsection 3.1.3). The edge length of each path is accumulated to obtain the total path length. For path P that has n edges $\overline{e}$, the total path length is defined as:

$$L = \sum_{i=1}^{n} |\overline{e}_i|$$

Where $|e|$ is the length of edge $\overline{e}$ and n is the length of a trajectory. See Figure 5. The result is normalized by the maximum path length L_{max} and stored in the GPU's vertex buffer. The vertex buffer is used for both color coding and the path filtering techniques.

3.1.5. Filtering Based on Total Path Curvature

Membrane curvature plays a key role in several biological functions and can influence the dynamic of proteins embedded in it Quemeneur et al [QSR*14]. Computational biologists are creating models to better understand the membrane curvature and how it can be induced by proteins West et al [WBB*16]. There are only few tools available to analyze curvature in membrane models Gapsys et al [GdGB13]. None of them allow the visualizing of the curvature for each lipid path. Curvature has multiple definitions the i) amount of deviation of an arc from a tangent line; ii) rate of change of the tangent direction; iii) reciprocal of the radius of the osculating circle; iv) an element of area of circular image/element of arclength Goldman [Gol05]. We compute the total path curvature by utilizing the winding-angle method presented by Sadarjoen and Post [SP00]. However, we use positive rotation for both a counterclockwise and a clockwise-rotating curve. For path P that has $n-1$ angles α the total path curvature of P is derived:

$$\alpha_w = \sum_{i=0}^{n} |\alpha_i|$$

Where $|\alpha|$ is the absolute value of α and α is the angle formed by two edges e. See Figure 5. Then the result is normalized by the maximum path curvature $\alpha_{w_{max}}$ before it is stored in the GPU's vertex buffer. The curvature of the trajectories can be represented by mapping the user selected color map to the curvature value. However, the user is able to set the filtering option to reduce the focus data via the range slider control.

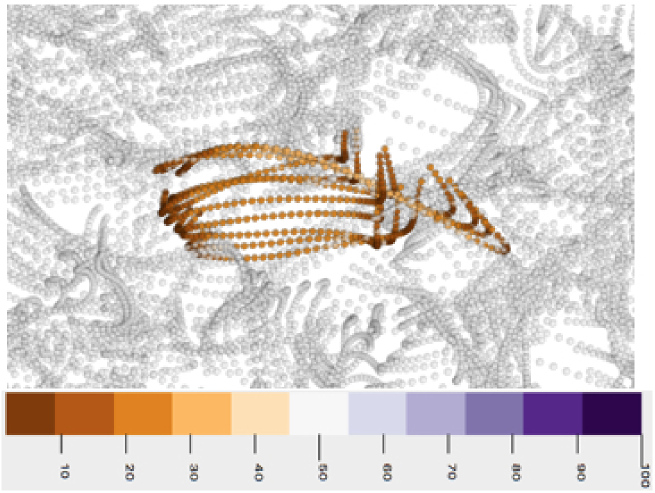

Figure 6: The color of atom reflects local edge length in 3D. Focus + context reduces the data of interest.

3.1.6. Filtering Based on Total Normalized Path Curvature

Another approach for investigating the trajectories based on their curvature is by computing the total normalized path curvature. We use the results in section 3.1.4 and section 3.1.5 to get the total normalized path curvature. For each path P the total normalized path curvature α_N is derived by dividing the path total curvature α_w by the total path length L:

$$\alpha_N = \frac{\alpha_w}{L}$$

The result is stored in the GPU's vertex buffer to be used for both filtering and color mapping.

3.1.7. Filtering Based on Depth of Path Starting Point

In order to derive helpful observations based on molecule dynamics in 3D space, we provide the user with a depth filtering technique based on the z component of the trajectories starting point p_0. The user is able to de-emphasize trajectories that are out of depth of interest by defining a minimum and maximum value of z . The depth filtering works alongside the other filtering and visualization techniques. See Figure 7. Membranes are deformable systems and can accommodate very curvy shapes. This filtering is useful to highlight areas of equivalent z value and then focus user attention on group of molecules gathered in the bottom or at the top of the membrane shape.

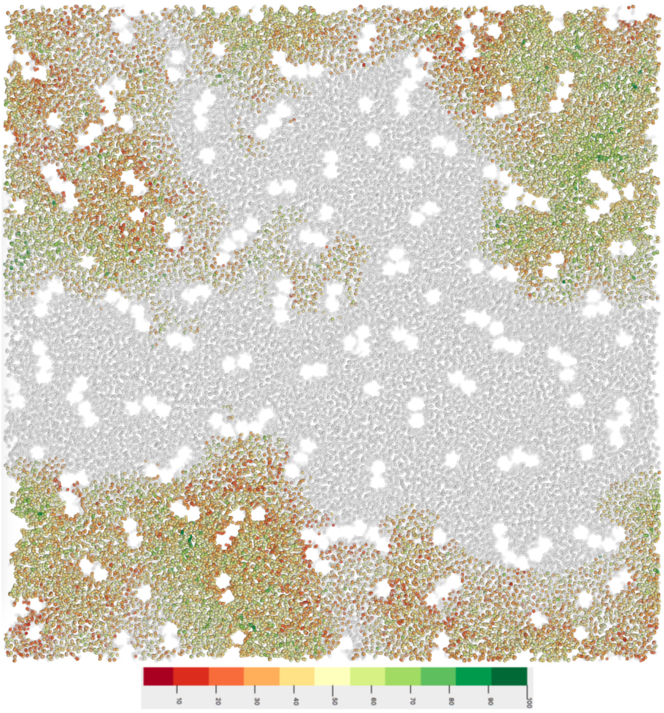

Figure 7: The z Depth filtering technique (min z =6.3 nano and max z =7.5 nano). Color is mapped to the total path length of the atom. The gray color indicates atoms outside the user-defined range.

3.1.8. Dual Space Representation

Projecting membrane properties onto a plane is a common simplification made by computational biologists to analyze their models (West et al. [WBB*16], Falck et al. [FRKV08], Lin et al. [LLG15]). Nevertheless, the membranes are 3D objects which are deformable. Assessing the differences between 2D and 3D rendering may help users choosing the most accurate depiction for the subsequent analyses.
The MolPathFinder provides the user with two spatial representations in order to study the data in 2D and 3D space. We explore a novel interactive 2D and 3D comparison visualization. The user can switch between the 2D and the 3D views. However, the novelty of our dual representation is that the user is able to compare between the properties of the trajectories in 2D and 3D simultaneously (see the supplementary video).

From the overview representation, the user gains insight into the global molecule dynamics based on the magnitude of the attributes of paths in 2D and 3D. See Figure 8. The user is able investigate interesting paths and visually compare them. First the trajectories are rendered in the 3D space. Next, by updating the same frame buffer, the trajectories are projected onto a 2D plane. For each point in the 2D path we project the z component onto the 2D plane. The path is projected onto the xy plane in such a way it has the same starting point as its dual in the 3D space. The representation shows the trajectories in the 3D and 2D simultaneously. The projected paths can be rendered on a single sliding plane or on a local plane.

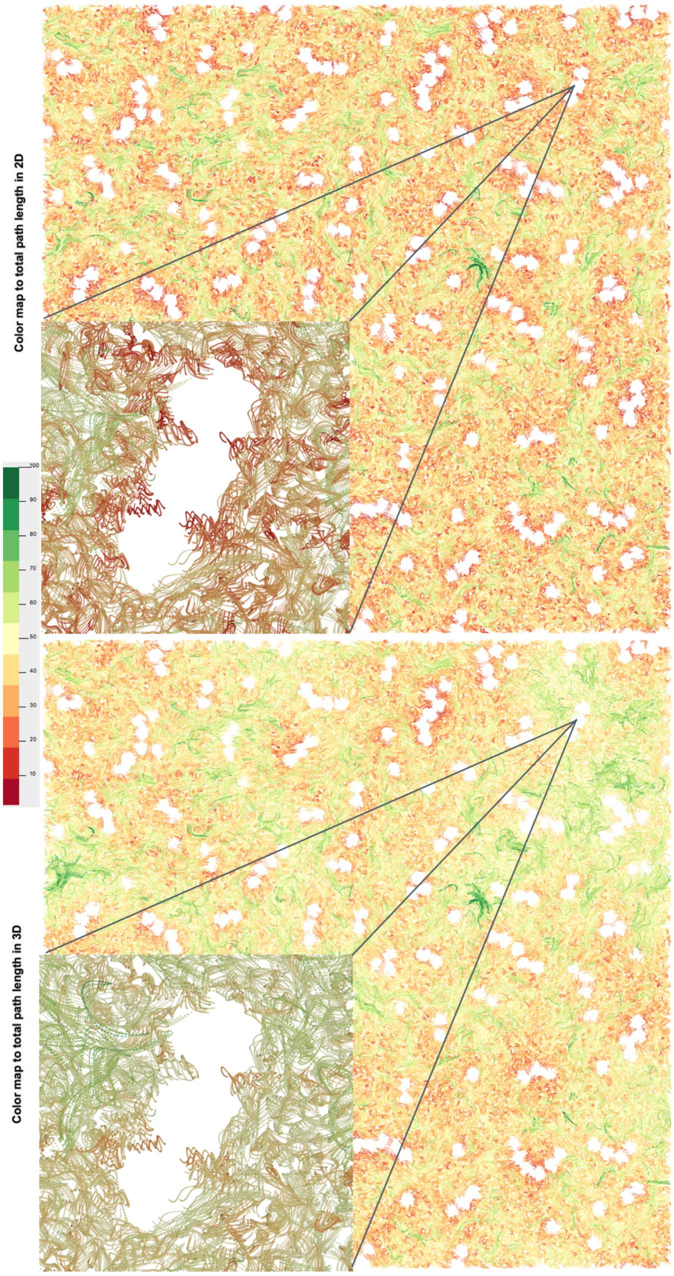
Color map to total path length in 2D
100
90
80
70
60
50
40
30
20
10
Color map to total path length in 3D

Figure 8: An overview comparison between total path length on 2D and 3D. Color is mapped to total length in 2D (top) and to total length in 3D (bottom). The comparison reveals a disadvantage of relying on a 2D representation. The total path length of atoms around proteins seems shorter in 2D view while it is not in 3D.

4. Experimental Results

Our experiments are carried out through analyzing and visualizing a real, time-dependent MDS data set. The size of the trajectory file is 76 MB. The trajectories involve the evaluation of 242,790 atoms over 51 ns. The dataset provided by our collaborators was creating to extend previous works to model bacterial membrane (Rassam et al. [RCB*15], Goose and Sansom [GS13]). To do so, our collaborators have designed very large models to address the questions of protein clusters at the mesoscale Chavent et al [CDS16]. The main goal of the dataset presented in this article is to understand the influence of such clusters on the lipid dynamics. The software is tested on Mac OS X El Capitan with a 4 GHz Intel Core i7 processor, 16 GB 1600 MHz DDR3 memory, and an AMD Radeon R9 M295X 4096 MB graphics card.

The Open Graphics Library 4.1 (OpenGL) and GL Shading Language 410 (GLSL) are utilized for computing and rendering the trajectory data. We use two shader programs (compute and render) to separate the computation from the rendering process. The former is used to update the color buffers based on the user requirements and the computational result. The second is used to render the data exploiting the updated color buffers. The frame rate is calculated while rendering the entire dataset. The number of rendered frames per second varies based on the selected representation approach. The option of rendering the atom as spheres results in 7 frames per second (FPS) and the path rendering option results in 10 FPS whereas the combination approach i.e. rendering atoms and their paths synchronously results in 5 FPS.

Observations The following observations were made as a result of utilizing the novel dual representation approach. By mapping the length of paths to color in 2D and 3D spaces. Figure 8 illustrates that the 2D representation conveys different information about the length of paths around proteins. The length of the paths around proteins in the 2D view seems to be short even though a lipids trajectory takes longer journey around the proteins in the 3D space. Therefore, 3D view provides us with good understanding of the behavior of the data.

With our approach, it appears very clearly that the dynamic of lipids is more important in the area devoid of protein (see Figure 3). Conversely, the path length are relatively shorter in the vicinity of proteins showing that the protein aggregates may hamper lipids dynamics.

Domain Expert Feedback This work has been developed in close collaboration with Dr. Chavent Matthieu from the Department of Biochemistry in Oxford university since the start of this project in January 2015. MolPathFinder is our first work and it is designed to fulfil the initial requirements of our collaborator. The tool video demo is available online (`https://vimeo.com/177758779`). The feedback shows positive reactions to our visualization software. The following is feedback from our domain expert collaboration:

"First of all, the software is very useful as, currently, no other program allows users to decipher the complex movements of lipids molecules from MD simulations. This is a very hot topic as it is now possible, using MD simulations, to model larger and larger systems (Ingólfsson et al. [IAPM16], Perilla et al. [PGC*15]) but the analysis and visualization of these systems is now becoming a bottleneck.

The visualization is really the main breakthrough of this program where it can help the user to understand in a glance the main movements of lipids molecules both at the nano-scale (i.e. zoom in: molecular level to see the movement of each lipid) and at the mesoscale (zoom out: global movement of lipids such as currents). So far only a recent work has begun to develop this type of rendering Chavent et al. [CRG*14] and there is a completely new field of research to develop around this thematic.

The GUI is well defined so that even a non-expert can easily interact with the data very instinctively. There are not too many windows and the labels are self-explanatory so that it is easy to quickly access and analyse the ones you want. There are several filtering functions that help the user to quickly focus on a specific area of interest.

Finally, the filtering in function of lipid path length, curvature, vorticity, etc.. is clearly new and a very valuable addition to this program. It was previously impossible to grasp this type of information in such an easy way."

Conclusions and Future Work Visualizing time-dependent trajectory data has received great attention over the past decades. A variety of visualization tools are used to investigate the MDS data at different resolutions. Our tool focuses on visualizing time-dependent trajectory MDS data at the atomic level. It provides the user with 2D and 3D representations and a number of different visualization techniques. The user is able to render the entire set of trajectories or a sub-set of the data. A variety of filtering techniques is designed to reduce the focus of the rendered data based on the user chosen-trajectory attributes. The dual representation helps to compare trajectories in 2D and 3D. Focus + context and zooming are used to explore the local behavior of the molecule dynamics.

This version of MolPathFinder relies on OpenGL to perform the computation task. However, other tools such as OpenCL and CUDA are designed for these purposes and they are a good candidates for future work. Our future work will involve three aspects: 1) handling higher resolution trajectory data, 2) utilizing quantification and a 2D graph, 3) introducing a new visualization techniques.

Acknowledgements We would like to thank the Ministry of Education of Saudi Arabia and the Saudi Cultural Bureau in London for their financial support on this project. We would also like to thank the Department of Computer Science at Swansea University for their support. M. C. is funded by the Wellcome Trust. ARCHER supercomputer was used to perform part of the simulations through a project funded by EPSRC-HECBiosim. Finally, we would like to thank Liam McNabb and Richard Roberts for help with proofreading the paper, and Joss Whittle for his advice on GPU techniques.

References

[ANG*10] APAJALAHTI T., NIEMELÄ P., GOVINDAN P. N., MIETTINEN M. S., SALONEN E., MARRINK S.-J., VATTULAINEN I.: Concerted diffusion of lipids in raft-like membranes. *Faraday discussions 144* (2010), 411–430. 4

[BGB*08] BIDMON K., GROTTEL S., BÖS F., PLEISS J., ERTL T.: Visual abstractions of solvent pathlines near protein cavities. *Computer Graphics Forum 27*, 3 (2008), 935–942. 2

[CDS16] CHAVENT M., DUNCAN A. L., SANSOM M. S.: Molecular dynamics simulations of membrane proteins and their interactions: from nanoscale to mesoscale. *Current Opinion in Structural Biology 40* (2016), 8–16. 1, 2, 7

[CLK*11] Chavent M., Lévy B., Krone M., Bidmon K., Nominé J.-P., Ertl T., Baaden M.: Gpu-powered tools boost molecular visualization. *Briefings in Bioinformatics* (2011), bbq089. 1, 2

[Col09] ColorBrewer: Colorbrewer 2.0., 2009. URL: http://colorbrewer2.org/. 3

[CRG*14] Chavent M., Reddy T., Goose J., Dahl A. C. E., Stone J. E., Jobard B., Sansom M. S.: Methodologies for the analysis of instantaneous lipid diffusion in md simulations of large membrane systems. *Faraday discussions 169* (2014), 455–475. 1, 4, 7

[DeL02] DeLano W. L.: Pymol: An open-source molecular graphics tool. *CCP4 Newsletter On Protein Crystallography 40* (2002), 82–92. 1, 2

[EKK*14] Ertl T., Krone M., Kesselheim S., Scharnowski K., Reina G., Holm C.: Visual analysis for space–time aggregation of biomolecular simulations. *Faraday discussions 169* (2014), 167–178. 2

[FRKV08] Falck E., Róg T., Karttunen M., Vattulainen I.: Lateral diffusion in lipid membranes through collective flows. *Journal of the American Chemical Society 130*, 1 (2008), 44–45. 6

[GdGB13] Gapsys V., de Groot B. L., Briones R.: Computational analysis of local membrane properties. *Journal of computer-aided molecular design 27*, 10 (2013), 845–858. 5

[GKM*15] Grottel S., Krone M., Müller C., Reina G., Ertl T.: Megamol - a prototyping framework for particle-based visualization. *IEEE transactions on visualization and computer graphics 21*, 2 (2015), 201–214. 1, 2

[Gol05] Goldman R.: Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design 22*, 7 (2005), 632–658. 5

[Goo05] Goodsell D. S.: Visual methods from atoms to cells. *Structure 13*, 3 (2005), 347–354. 2

[GRO09] GROMACS: Xtc library, 2009. URL: http://www.gromacs.org/Developer_Zone/Programming_Guide/XTC_Library. 3

[GS13] Goose J. E., Sansom M. S.: Reduced lateral mobility of lipids and proteins in crowded membranes. *PLoS Comput Biol 9*, 4 (2013), e1003033. 7

[HDS96] Humphrey W., Dalke A., Schulten K.: Vmd: visual molecular dynamics. *Journal of molecular graphics 14*, 1 (1996), 33–38. 1, 2

[HGB14] Hirst J. D., Glowacki D. R., Baaden M.: Molecular simulations and visualization: introduction and overview. *Faraday discussions 169* (2014), 9–22. 2

[HS16] Hedger G., Sansom M. S.: Lipid interaction sites on channels, transporters and receptors: recent insights from molecular dynamics simulations. *Biochimica et Biophysica Acta (BBA)-Biomembranes* (2016). 1

[IAPM16] Ingólfsson H. I., Arnarez C., Periole X., Marrink S. J.: Computational 'microscopy' of cellular membranes. *J Cell Sci 129*, 2 (2016), 257–268. 7

[JMS16] Javanainen M., Martinez-Seara H.: Efficient preparation and analysis of membrane and membrane protein systems. *Biochimica et Biophysica Acta (BBA)-Biomembranes* (2016). 2

[KKL*15] Kozlíková B., Krone M., Lindow N., Falk M., Baaden M., Baum D., Viola I., Parulek J., Hege H.: Visualization of molecular structure: The state of the art. *EuroVis* (2015). 2

[KP*90] Karplus M., Petsko G. A., et al.: Molecular dynamics simulations in biology. *Nature 347*, 6294 (1990), 631–639. 2

[Kri03] Krieger E.: Yasara, 2003. URL: www.yasara.org/. 2

[KV14] Krieger E., Vriend G.: Yasara view - molecular graphics for all devices - from smartphones to workstations. *Bioinformatics 30*, 20 (2014), 2981–2982. 2

[LBH11] Lindow N., Baum D., Hege H.-C.: Voronoi-based extraction and visualization of molecular paths. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2025–2034. 2

[LLC*12] Lipşa D. R., Laramee R. S., Cox S. J., Roberts J. C., Walker R., Borkin M. A., Pfister H.: Visualization for the physical sciences. 2317–2347. 2

[LLG15] Lin X., Li Z., Gorfe A. A.: Reversible effects of peptide concentration and lipid composition on h-ras lipid anchor clustering. *Biophysical journal 109*, 12 (2015), 2467–2470. 6

[LTDS*13] Lv Z., Tek A., Da Silva F., Empereur-Mot C., Chavent M., Baaden M.: Game on, science-how video game technology may help biologists tackle visualization challenges. *PloS one 8*, 3 (2013), e57990. 1, 2

[OGF*10] O'Donoghue S. I., Goodsell D. S., Frangakis A. S., Jossinet F., Laskowski R. A., Nilges M., Saibil H. R., Schafferhans A., Wade R. C., Westhof E., et al.: Visualization of macromolecular structures. *Nature methods 7* (2010), S42–S55. 2

[OGG*10] O'Donoghue S. I., Gavin A.-C., Gehlenborg N., Goodsell D. S., Hériché J.-K., Nielsen C. B., North C., Olson A. J., Procter J. B., Shattuck D. W., et al.: Visualizing biological data - now and in the future. *Nature methods 7* (2010), S2–S4. 2

[PGC*15] Perilla J. R., Goh B. C., Cassidy C. K., Liu B., Bernardi R. C., Rudack T., Yu H., Wu Z., Schulten K.: Molecular dynamics simulations of large macromolecular complexes. *Current opinion in structural biology 31* (2015), 64–74. 1, 2, 7

[PGH*04] 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. *Journal of computational chemistry 25*, 13 (2004), 1605–1612. 2

[QSR*14] Quemeneur F., Sigurdsson J. K., Renner M., Atzberger P. J., Bassereau P., Lacoste D.: Shape matters in protein mobility within membranes. *Proceedings of the National Academy of Sciences 111*, 14 (2014), 5083–5087. 5

[RCB*15] Rassam P., Copeland N. A., Birkholz O., Tóth C., Chavent M., Duncan A. L., Cross S. J., Housden N. G., Kaminska R., Seger U., et al.: Supramolecular assemblies underpin turnover of outer membrane proteins in bacteria. *Nature 523*, 7560 (2015), 333–336. 1, 7

[SP00] Sadarjoen I. A., Post F. H.: Detection, quantification, and tracking of vortices using streamline geometry. *Computers & Graphics 24*, 3 (2000), 333–341. 5

[VDSLH*05] Van Der Spoel D., Lindahl E., Hess B., Groenhof G., Mark A. E., Berendsen H. J.: Gromacs: fast, flexible, and free. *Journal of computational chemistry 26*, 16 (2005), 1701–1718. 2

[WBB*16] West A., Brummel B. E., Braun A. R., Rhoades E., Sachs J. N.: Membrane remodeling and mechanics: Experiments and simulations of α-synuclein. *Biochimica et Biophysica Acta (BBA)-Biomembranes 1858*, 7 (2016), 1594–1609. 5, 6