

Michael Wiebrands<sup>1</sup> / Chris J. Malajczuk<sup>2</sup> / Andrew J. Woods<sup>1</sup> / Andrew L. Rohl<sup>3</sup> / Ricardo L. Mancera<sup>4</sup>

# Molecular Dynamics Visualization (MDV): Stereoscopic 3D Display of Biomolecular Structure and Interactions Using the Unity Game Engine

<sup>1</sup> Curtin Hub for Immersive Visualization and eResearch (HIVE), Curtin University, Perth, WA, Australia

<sup>2</sup> School of Pharmacy and Biomedical Sciences, Curtin Health Innovation Research Centre and Curtin Institute for Computation, Curtin University, Perth, WA, Australia

<sup>3</sup> School of Molecular and Life Sciences and Curtin Institute for Computation, Curtin University, Perth WA, Australia

<sup>4</sup> School of Pharmacy and Biomedical Sciences, Curtin Health Innovation Research Centre and Curtin Institute for Computation, Curtin University, GPO Box U1987, Perth, WA 6845, Australia, E-mail: R.Mancera@curtin.edu.au

## Abstract:

Molecular graphics systems are visualization tools which, upon integration into a 3D immersive environment, provide a unique virtual reality experience for research and teaching of biomolecular structure, function and interactions. We have developed a molecular structure and dynamics application, the Molecular Dynamics Visualization tool, that uses the Unity game engine combined with large scale, multi-user, stereoscopic visualization systems to deliver an immersive display experience, particularly with a large cylindrical projection display. The application is structured to separate the biomolecular modeling and visualization systems. The biomolecular model loading and analysis system was developed as a stand-alone C# library and provides the foundation for the custom visualization system built in Unity. All visual models displayed within the tool are generated using Unity-based procedural mesh building routines. A 3D user interface was built to allow seamless dynamic interaction with the model while being viewed in 3D space. Biomolecular structure analysis and display capabilities are exemplified with a range of complex systems involving cell membranes, protein folding and lipid droplets.

**Keywords:** Multi-user cylindrical display, polygonal mesh, stereoscopic user interface, virtual reality

**DOI:** 10.1515/jib-2018-0010

**Received:** January 23, 2018; **Revised:** April 16, 2018; **Accepted:** May 9, 2018


## 1 Introduction

Molecular simulations are used to study the structure, function and interactions of biomolecules. In these simulations, molecular motion and forces are calculated with supercomputers over timescales ranging from picoseconds to microseconds. Knowledge of the position and velocity of every atom across time allows the prediction of multiple molecular and bulk properties. Molecular graphics systems are visualization tools that can greatly enhance the use of molecular simulations to study the behavior of biomolecules. Molecular mechanisms in biology can indeed be elucidated through a combination of computation and visualization.

The ability to display complex structures using a stereoscopic 3D display allows the layout of such structures to be understood much more easily than a conventional 2D display technique. The ability to display datasets in a large immersive display can further enhance interpretation and engagement. The use of immersive 3D display systems can aid researchers to analyse new or existing datasets, communicate ideas and concepts, and showcase findings with the wider community.

In 2013, Curtin University launched the Hub for Immersive Visualization and eResearch (HIVE) visualization facility [1]. The HIVE is a multi-user, multi-display facility which supports, encourages and enables the use of visualization technologies across a wide range of research projects across various disciplines. One of the displays at the HIVE is called the Cylinder and consists of a 3 m high × 8 m diameter screen with a 180° field-of-view from the apex. The screen is illuminated using three 1920 × 1200 DLP projectors warped and blended

**Ricardo L. Mancera** is the corresponding author.

 ©2018, Michael Wiebrands et al., published by De Gruyter.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.

to provide a continuous image across the screen. The system supports 3D display using liquid-crystal shutter (active) 3D glasses. The system can be successfully used with groups of up to 50 people.

In this work, we aimed to explore how the use of the HIVE Cylinder display could potentially enhance the visualization and interpretation of molecular structures – both static and dynamic – for research, teaching and communication purposes. Here we report the development of a molecular structure and dynamics visualization tool using the Unity cross-platform game engine to deliver an immersive display experience on large scale, multi-user, stereoscopic visualization systems. This application is, to our knowledge, the first 3D molecular visualization tool developed for large scale, multi-user stereoscopic visualization systems using the Unity game engine. It is also the first known tool to allow the loading and interaction of stereoscopic molecular visualizations in real-time through a 3D user interface.

## 2 Related Works

Interactive rendering of biomolecular structure and dynamics has always represented a visualization challenge due to the rapid loss of performance due to the constant changes of complex structures. Some approaches to solve this problem have aimed to align the hierarchical structure of proteins with a two-level approach, such that dynamic changes are applied to the protein backbone (high-level structure) and the geometry of amino acid residues (low-level structures) is generated on-the-fly for atom-by-atom rendering. This has been used to enable visualization of slow dynamics of large protein assemblies [2].

Other efforts have centered on developing virtual reality (VR) tools that are controlled by gesture recognition, such as the Molecular Rift, which was developed using the Unit 3D engine to work with the Oculus Rift headset [3]. Similar tools are being developed in combination with applications for the analysis of quantum mechanics and molecular mechanics calculations, such as in the case of the Caffeine molecular viewer [4].

One of the most well-known molecular graphics viewers, Visual Molecular Dynamics (VMD) [5], is a versatile desktop application for displaying, animating, and analyzing large biomolecular systems using 3D graphics and built-in scripting. VMD can render complex images to a range of different display devices, and it supports the use of head-mounted displays combining omni-directional stereoscopic visualization with ray-tracing rendering [6]. It also supports a cave automatic virtual environment (CAVE), a system which provides an immersive, stereoscopic environment with head tracking using projectors directed to between three and six walls of a room-sized cube [7]. Despite its versatility, we were unaware of a way of having VMD display content on the HIVE Cylinder display in 3D with the correct display geometry.

Graphics for the HIVE Cylinder display are usually rendered as a series of twelve vertical strips across the width of the curved display. Each strip is actually a stereoscopic strip such that for 12 strips there are 24 virtual cameras. This is necessarily an approximation of a curved display surface, and more cameras would improve the quality of the geometry; however, it appears that 12 strips strikes a good balance between accuracy of the geometry and rendering speed (which slows down with more cameras). Most applications cannot do anything other than a regular flat display so it can be difficult to find software which will render graphics correctly for the HIVE Cylinder display. At the beginning of this work, we already had a technique of correctly rendering graphics on the Cylinder display using the Unity development engine. Consequently, we decided to implement the visualization of molecular structures and simulations using the Unity game engine (<https://unity3d.com/>).

UnityMol is an existing Unity-based molecular visualization application [8] which exploits ray casting on GPUs by using the so-called HyperBalls representation, an improved ball-and-stick representation that replaces tubes, linking atom spheres by hyperboloids to smoothly connect them [9]. This approach is well suited to represent dynamical phenomena such as the evolution of non-covalent bonds. It was built and optimized specifically for desktop workstations and not for stereoscopic or multi-projection displays. The menu and control system are monitor-based and do not render in stereo. Application performance is variable with quite low frame rates at times, which would require substantial modifications to perform adequately with MiddleVR, the Unity middleware used for the Cylinder display.

## 3 Implementation

We developed the Molecular Dynamics Visualization (MDV) tool using the Unity game engine (<https://unity3d.com/>). We use the MiddleVR plugin for Unity that allows visualization to run in stereoscopic mode on a multi-projector display setup. In particular, we aimed to display biomolecular structures on the large cylindrical projection display in the Curtin HIVE. The application was developed to also run on Windows PC desktops for local demonstration and testing purposes.

The application is structured to separate the biomolecular modeling and visualization systems. The biomolecular model loading and analysis code base was developed as a stand-alone C# library, which is the default language for Unity, thus providing the foundation for the custom visualization system built in Unity. The application was built utilising a Model View Controller (MVC) architecture. The model is a custom DLL separate from Unity and the view and controller exist within the Unity scripting code. Separating out the model code from the Unity C# code allowed for simpler development of a test suite to provide independent validation of the molecular model, as this is often quite difficult to do in Unity itself.

All visual models displayed within the tool are generated using Unity-based procedural mesh building routines. A 3D user interface was built to allow seamless dynamic interaction with the model while being viewed in 3D space. Protein secondary structure visualization support is in part facilitated by the integration of Stride, a publicly available secondary structure analysis tool. The MDV tool performs a secondary structure analysis of proteins using Stride in real-time to dynamically update the secondary structure visualization when viewing an animated molecular dynamics simulation trajectory.

The MDV tool provides a high-resolution 3D representation of biomolecular structure, function and interactions. It supports visualization of the most common protein structure file format [Protein Data Bank or PDB (<https://www.rcsb.org/>)] as well as file types generated by the widely used molecular dynamics simulation software package GROMACS [10].

### 3.1 Model

The model of the application was developed in Microsoft Visual Studio as a stand-alone C# library which was loaded into Unity as a DLL. The model consists of data parsers (model loaders) and molecular model classes which aid the extraction and analysis of molecular model data and the return of that same data in formats usable by the Unity visualization routines.

The data parsers accept the local file path of a molecular structure or molecular dynamics trajectory input files and return the information from the file as a set of molecular model classes. Parsers have been written to support the loading of GRO, PDB and XYZ structure formats and XTC and DCD trajectory formats, allowing the use of output files produced by common molecular modeling programs and molecular dynamics simulation packages such as GROMACS, LAMMPS and NAMD. The parsers assume a protein structure and will fail on other structures present in the input files. All the parsers are custom apart from the XTC parser. The core of the XTC trajectory file parser code was taken from the open source VMD XTC C file parsing module and converted into a C# class.

The model parsers support the extraction, analysis and identification of primary structure information including atoms, bonds, amino acids residues and amino acid chains. The model also supports the extraction and identification of protein secondary structure information including three-turn helices, alpha helices, five-turn helices, hydrogen-bonded turns, beta sheets, beta bridges, bends and coils.

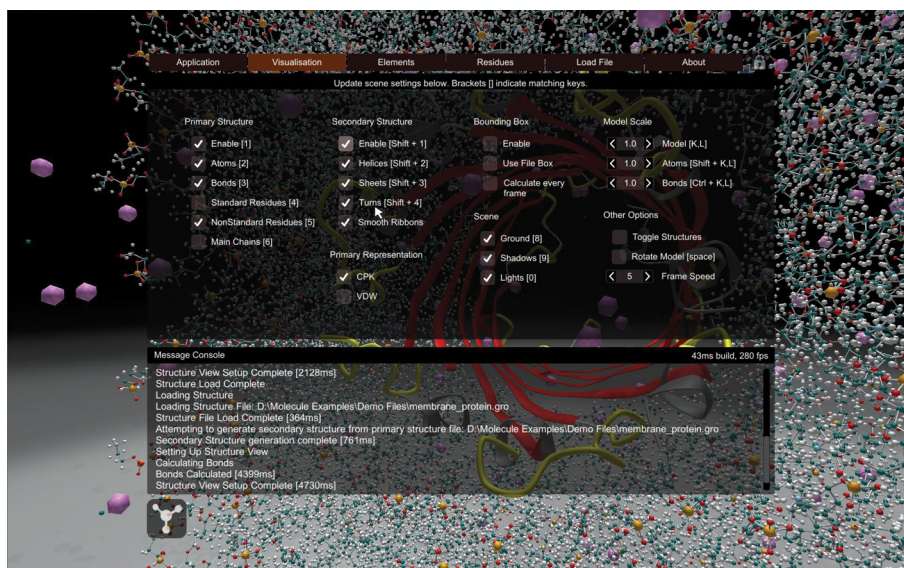
Protein secondary structure information is available for the base structure and for every frame of the molecular model trajectory. Given the trajectory frame count can be very large, the secondary structure information is not extracted by a parser but instead processed on demand. Secondary structure analysis is performed by a separate executable implementing Stride [11] (<http://webclu.bio.wzw.tum.de/stride/>). When retrieving secondary structure information, the model generates a temporary custom PDB file based on the data stored in the molecular model classes, passes it to the executable, and then parses the results back into a molecular model class for use by the visualization routines.

### 3.2 View Controller

The controller for the application is a central script attached to a scene manager GameObject in Unity. It loads the molecular model from the model DLL, passes the molecular model to the visualization routines to display on the screen and accepts inputs from the in-scene user interface to manipulate the on-screen view of the molecular model.

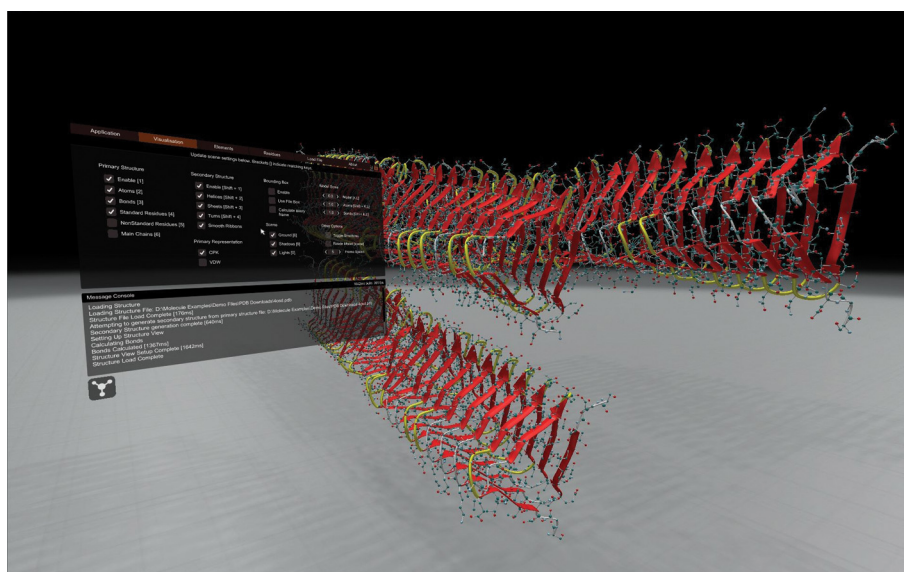
### 3.3 User Interface

The user interface (UI) has been designed to work both on a flat screen monitor and in stereo on a large screen cylindrical display. It works similarly to a standard overlaid 2D menu system and uses standard Unity UI components and a mouse pointer to select parts of the UI (Figure 1).



**Figure 1:** User interface in 2D on desktop PC.

Stereo applications do not directly support screen space UIs or Windows mouse pointers. Consequently, the interface is instead mapped to a 3D plane in world space and a virtual mouse pointer is positioned in 3D space just above the plane that the UI is mapped to (Figure 2). The virtual pointer responds to mouse movement but is locked to the surface and bounds of the UI plane. Left click events from the pointer interact with the UI by ray casting from the scene camera through the pointer to the UI plane and triggering a Unity UI event for the underlying UI component.



**Figure 2:** User interface in 3D on stereoscopic display.

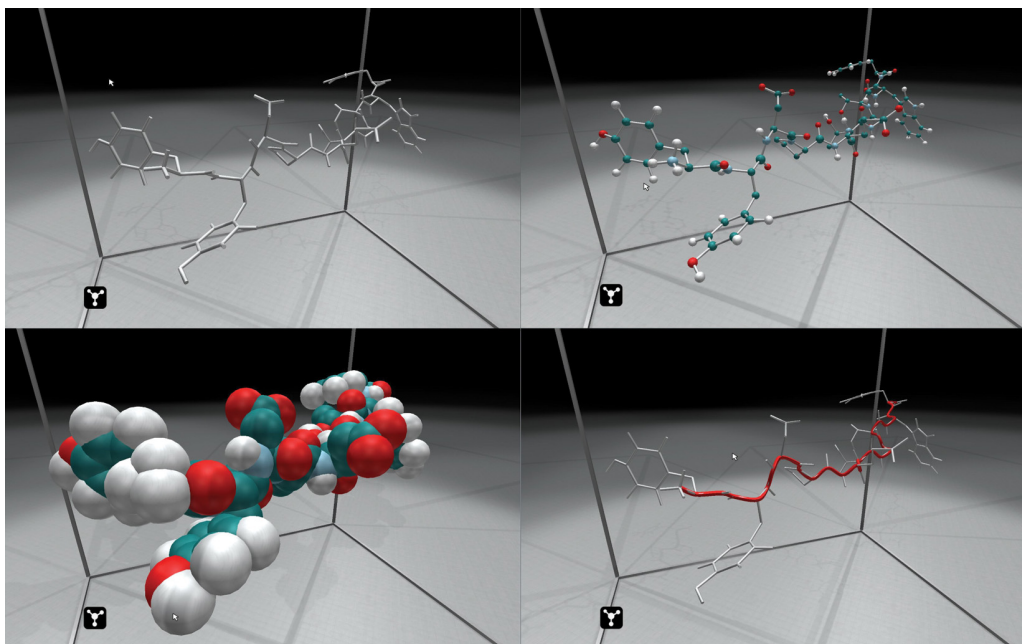
When the UI plane is orthogonal to the forward camera projection on a 2D monitor, the UI appears to function identically to a standard screen space user interface. In stereo, the UI appears to float in front of the user in the 3D space but still retains the same usability and expected function, creating a unified control scheme between both the 2D and stereo formats of the application. A short movie illustrating this use of the user interface on various molecular systems can be found online [12].

Also, the existing molecular visualization applications that provide stereo support generally provide a limited stereoscopic interface, requiring dropping out of stereo and at times restarts of the application to load and manipulate the view in stereo. This solution provides full input file loading and model manipulation while viewing in stereo, making the solution much simpler to use. This was particularly important when viewing the molecular models on a large screen in a poorly lit room.

### 3.4 Molecular Visualization

The application supports visualization of the primary and secondary structure of the molecular model as objects in 3D space. A 3D object is most commonly visualized with a polygon mesh overlaid with one or more textures. In Unity this is performed by adding a mesh and material to a GameObject and then positioning the GameObject in 3D space.

The primary structure of the molecular model consists of simple spherical atoms and cylindrical bonds connecting the atoms and can be represented by simple spherical and cylindrical meshes overlaid by textures coloring the atoms to match their element types. Various combinations and sizing of these atom and bond meshes in the application allow the display of three commonly-used, primary structure representations: CPK, space-filling (VDW) and bond only (Figure 3).

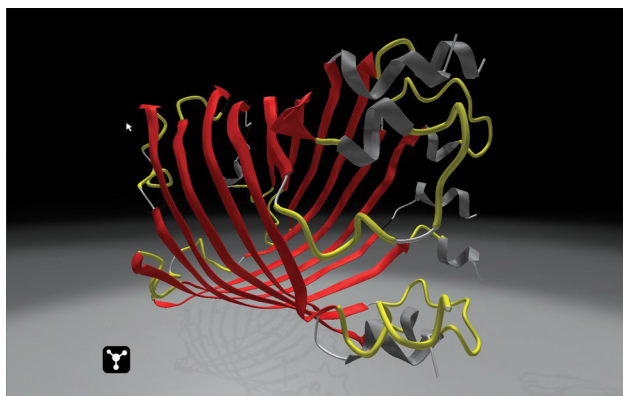


**Figure 3:** Primary representations using simple meshes: bond (top left), CPK (top right) and space-filling (bottom left). Bottom right image shows procedural mesh of protein chain overlaid on a bond representation.

In the first iteration of the application, the molecular model atoms and bonds were represented with a single GameObject for each, with the intention to manipulate the individual atom and bond positioning to animate the molecular model. Unfortunately, Unity performance is significantly degraded once the number of GameObjects was increased to the tens of thousands, making this approach infeasible since the molecular models can have atom counts in the hundreds of thousands. Fortunately, during the development of this project, Unity implemented GPU instancing, which provides significant performance improvements when rendering multiple copies of the same mesh at once. Unfortunately, the performance was still not adequate.

The final solution instead merges all the individual simple atom and bond meshes into a small number of complex meshes using custom-built mesh merge routines. For each element type in the model, the atom meshes for that element are combined into a single mesh and the texture associated with that element type applied to the mesh. With this method, the number of meshes, and associated GameObjects is reduced a thousand-fold and the performance of the application, as measured by frame rate, is improved on average by approximately ten times. The only downside to this solution is that individual atoms and bonds in the molecular model view cannot be moved independently without recreating the whole mesh. As such the animation for the model trajectory cannot easily be interpolated and currently lacks smooth transitions.

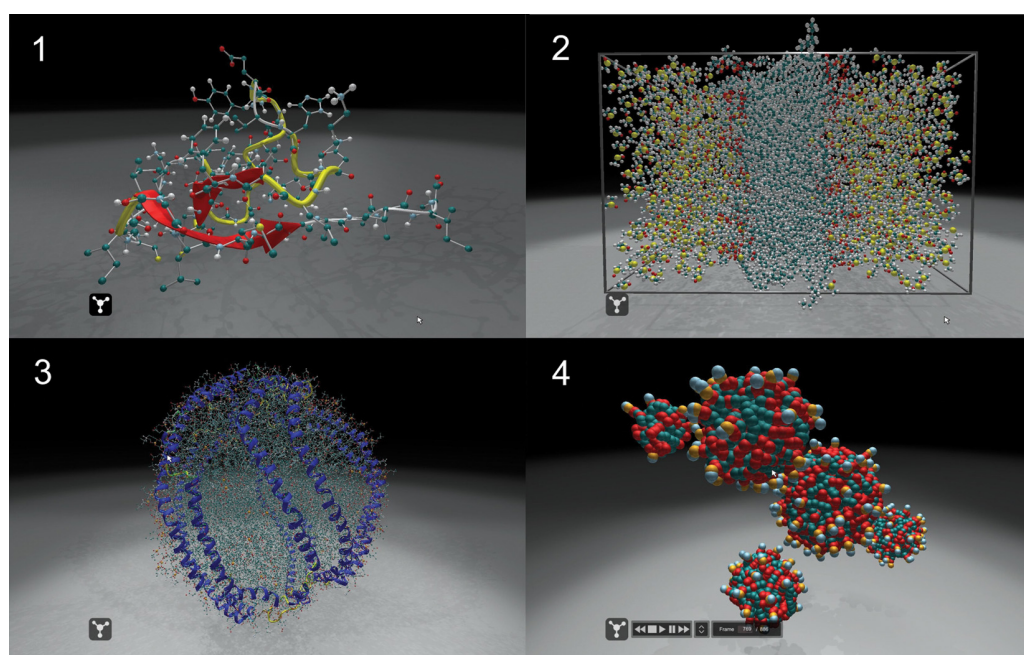
The protein secondary structure of the molecular model is defined by a mapping of secondary structure type along the backbone of the protein chain, with the type being defined for each amino acid in the chain sequence. As such, the secondary structure can be visualized as a single long tubular structure that deforms into different shapes representing the structure types. In the application, this was achieved by scripting a custom tube mesh routine that generates a tube of arbitrary size and deforms into a flat ribbon as needed to represent the alpha helices and beta sheets within the secondary structure representation (Figure 4).



**Figure 4:** Three intertwined tube meshes representing the secondary structure of a membrane protein.

## 4 Application

We tested the capability of this tool with four systems of varied complexity: (1) the conformational dynamics of an intrinsically disordered small protein characterized through a molecular dynamics simulation (Figure 5.1); (2) the interactions of a model cell membrane with an aqueous solution of small molecules (Figure 5.2); (3) the static structure of a representative model of a high-density lipoprotein particle (Figure 5.3); and (4) the coarse-grained molecular dynamics simulation of the spontaneous formation of nanoscale lipid-droplets (Figure 5.4).



**Figure 5:** Biomolecular systems used to test the capability of the application: (1) the conformational dynamics of an intrinsically disordered protein; (2) the interactions of a model cell membrane with an aqueous solution of small molecules; (3) the structure of a high-density lipoprotein particle; and (4) the spontaneous aggregation of nanoscale lipid droplets.

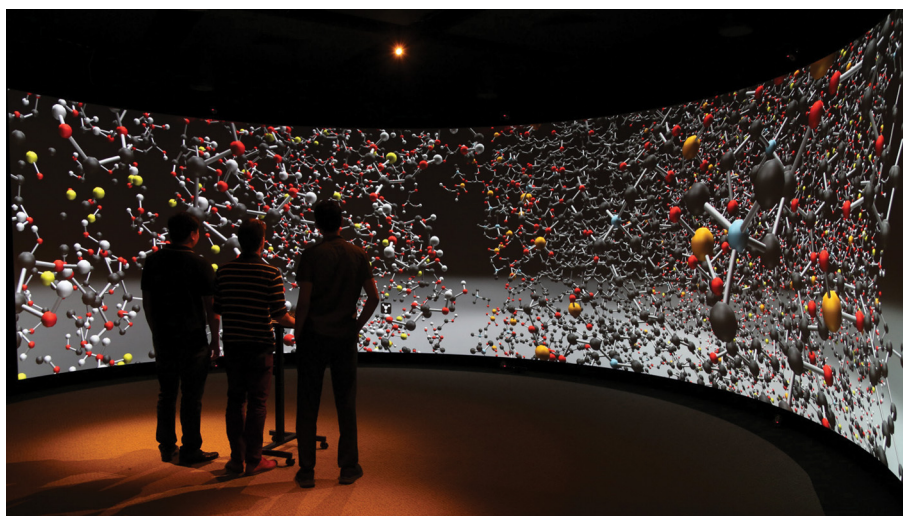
Not all of these test cases may necessarily warrant the use of a large, cylindrical projection display. However each one was chosen to develop specific capabilities. Most of these test cases are illustrated in a short online movie [12]. System 1 was used to test the dynamic animation of changes in secondary structure of a protein while progressing through a molecular dynamics simulation trajectory. This system demonstrates the capability of the application to retrieve the secondary structure information from Stride and reconstruct the secondary structure mesh dozens of times per second. System 2 was used to test the ability to highlight and track a small molecule passing through a phospholipid model cell membrane. System 3 tested the performance of the application when displaying a static complex system. The model consists of 34,867 atoms and 35,162 bonds, displaying at approximately 80 fps on a large multi projector cylindrical display (~700 fps on high end desktop PC). System 4 was used to test the visualization of a molecular dynamics simulation of a system displayed using a space-filling representation.

## 5 Discussion

Tools for visualizing biomolecular systems are common. Unfortunately, current visualization tools are often incompatible with large screen multiuser display formats. For the cylindrical displays such as the one at our institution, the calculation and projection of dozens of virtual cameras for the visualization scene initially had a severe impact on performance, with stereoscopic 3D applications generally running an order of magnitude slower than on high end desktop PCs. For this application, the custom software routines developed to create highly optimized scene meshes dramatically increased the performance of the visualizations and demonstrated a compelling way to render detailed visualizations at acceptable speeds on large screen displays.

Stereoscopic rendering makes traditional user interface options unworkable on large screen stereoscopic displays. This problem also exists in modern VR software development, where existing control interfaces need quite radical redevelopment to work in stereo. For purposes where the user is stationary, the mouse based virtual user interface developed for this application works very well. Mapping a traditional 2D interface into 3D allows those who might be unfamiliar with arbitrary 3D control interfaces to work with a control scheme that is immediately more familiar and intuitive. It also allows an easy transition for the user who is using the application on both the desktop and in stereo on the large screen.

The HIVE Cylinder display can be used with small or large audiences up to 50 people, all wearing 3D glasses, in a standing configuration for smaller groups or a seated arrangement for larger groups (see Figure 6). The wrap around 3D nature of the display provides the audience with a very immersive experience of the molecular simulation which not only increases impact but also aids understanding of complex structures and systems. We often explain that the HIVE Cylinder display is like Virtual Reality (VR) using a Head Mounted Display (HMD) but this system allows group collaborative experiences in a much easier way than using multiple headsets. The use of a large display with the audience spread over a large area in front of the display does incur some image distortions for people away from the “sweet-spot”; however, other work has found that for a curved display such as the HIVE Cylinder display, those distortions are gradual and mostly acceptable [13].



**Figure 6:** MDV running on the HIVE Cylinder display.

The biomolecular systems utilized to demonstrate real world application prove the visualization tool as capable and compelling. Since the recent completion of the MDV tool, it has been widely used for educational, research and promotional purposes making it a valuable tool for researchers, the visualization centre and the university.

### Acknowledgements

CJM gratefully acknowledges the Australian Government for an Australian Postgraduate Award (APA) scholarship. Some of this work was conducted as a HIVE summer research internship project supported by the Curtin HIVE and the Curtin Institute for Computation (CIC).

**Conflict of interest statement:** Authors state no conflict of interest. All authors have read the journal’s Publication ethics and publication malpractice statement available at the journal’s website and hereby confirm that they comply with all its parts applicable to the present scientific work.

## References

- [1] Woods AJ, Datta S, Bourke P, Hollick J. The design, install and operation of a multi-user, multi-display visualisation facility. Manuscript in preparation.
- [2] Lampe OD, Viola M, Reuter N, Hauser H. Two-level approach to efficient visualization of protein dynamics. *IEEE Trans Vis Comput Graph*. 2007;13:1616–23.
- [3] Norrby M, Grebner C, Eriksson J, Boström J. Molecular rift: virtual reality for drug designers. *J Chem Inform Model*. 2015;55:2475–84.
- [4] Salvadori A, Del Frate G, Pagliai M, Mancini G, Barone V. Immersive virtual reality in computational chemistry: applications to the analysis of QM and MM data. *Int J Quantum Chem*. 2016;116:1731–46.
- [5] Humphrey W, Dalke A, Schulten K. VMD: visual molecular dynamics. *J Mol Graph*. 1996;14:33–8.
- [6] Stone JE, Sherman WR, Schulten K. Immersive molecular visualization with omnidirectional stereoscopic ray tracing and remove rendering. In: *IEEE international Parallel and Distributed Processing Symposium Workshop (IPDPSW)* 2016:1048–57.
- [7] Cruz-Neira C, Sandin DJ, DeFanti TA, Kenyon RV, Hart JC. [The CAVE: audio visual experience automatic virtual environment](#). *Commun ACM*. 1992;35:64–72.
- [8] 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*. 2013;8:e57990.
- [9] Chavent M, Vanel A, Tek A, Levy B, Robert S, Raffin B, et al. GPU-accelerated atom and dynamic bond visualization using hyperballs: a unified algorithm for balls, sticks, and hyperboloids. *J Comput Chem*. 2011;32:2924–35.
- [10] Abraham M, Murtola T, Schulz R, Pall S, Smith JC, Hess B, et al. GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*. 2015;1–2:19–25.
- [11] Frishman D, Argos P. Knowledge-based protein secondary structure assignment. *Proteins Struct Funct Genet*. 1995;23:566–79.
- [12] Molecular Dynamics Visualisation Tool. <https://youtu.be/5srmfuhlrb5>
- [13] Woods AJ, Bourke P, Hollick J, Helliwell J. Image Distortions in Large-Scale Immersive 3D Display Systems – Cylinder and Wedge Displays. In: *Stereoscopic Displays and Applications XXVIII*. 2017. (Presentation Only). Presentation Video: <https://www.youtube.com/watch?v=OF7mlwaiEc4>.