Received: 18 April 2016 | Revised: 21 June 2016 | Accepted: 24 June 2016

DOI 10.1002/qua.25207

SOFTWARE NEWS & UPDATES

WILEY International Journal of QUANTUM CHEMISTRY

Immersive virtual reality in computational chemistry: Applications to the analysis of QM and MM data

Andrea Salvadori | Gianluca Del Frate | Marco Pagliai | Giordano Mancini | Vincenzo Barone

Scuola Normale Superiore, Piazza dei Cavalieri 7, Pisa I-56126, Italy

Correspondence
Giordano Mancini, Scuola Normale Superiore, Classe di Scienze, Piazza dei Cavalieri 7, Pisa, Italy.
Email: giordano.mancini@sns.it

Funding Information
Support from the Italian MIUR (FIRB 2012 Project no. RBFR12ETL5) is acknowledged. The research leading to the results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. [320951].

Abstract

The role of Virtual Reality (VR) tools in molecular sciences is analyzed in this contribution through the presentation of the Caffeine software to the quantum chemistry community. Caffeine, developed at Scuola Normale Superiore, is specifically tailored for molecular representation and data visualization with VR systems, such as VR theaters and helmets. Usefulness and advantages that can be gained by exploiting VR are here reported, considering few examples specifically selected to illustrate different level of theory and molecular representation.

KEYWORDS
data interaction, molecular viewers, virtual reality

1 | INTRODUCTION

A detailed, yet compact, representation of molecular structures, together with the inclusion of related properties in formulas and graphs, has always been at the heart of chemistry. Representation plays a key role in the whole discovery process, conveying information to human inspectors, relying on human pattern recognition, and suggesting innovative points of investigation and new, previously unexplored scenarios.[1] From a theoretical chemistry perspective, without molecular graphics, the sheer amount of information provided by current computational power would rather hinder true knowledge acquisition.[2] The importance of molecular graphics in chemistry is demonstrated by its leading role in the adoption of advances in computer graphics for scientific visualization.[3] The evolution of computer technologies for three dimensional immersive virtual reality (IVR) allows nowadays to achieve a further evolution in data representation and visualization.[4] In fact, it is now possible to create 3D virtual environments that extend users perception and increase researchers ability to quickly tackle massive amounts of data coming from multiple and different sources. Within such systems, users can directly interact with visualized data (by means of dedicated devices) in a more natural and friendly way than that achievable on desktop systems with mouse and keyboard.[5,6] IVR technologies include a large panel of devices, from cheap consumer grade ones to very costly specialized hardware. In the first category, we can mention interactive sensors like the *Microsoft Kinect*[7] and the *Leap Motion*,[8] current generation immersive helmets such as the *Oculus Rift*[9] and the *Vive* from *HTC* and *Valve*,[10] or force-feedback devices like the *Novint Falcon 3D Touch controller*.[11] The second category instead includes virtual theaters, such as the Cave Automatic Virtual Environment (CAVE),[12,13] equipped with high-precision tracking sensors and driven by one or more powerful workstations.

With the aim of enabling the employment of different platforms, ranging from desktop computers to more expensive IVR installations, we are developing a new molecular viewer, called *Caffeine*.[14] Beside motivations of performance and the aim to exploit state of the art technologies, one of the reasons for developing a new molecular viewer was not to force IVR features in an application designed within a different scope. Rather, we tried to design the software in such a way that the transition from a familiar 2D desktop environment to an IVR one would be as smooth and easy as possible. A specific feature to which we payed great attention in the development is coupling the visualization of molecular structures with the plotting of numerical data,

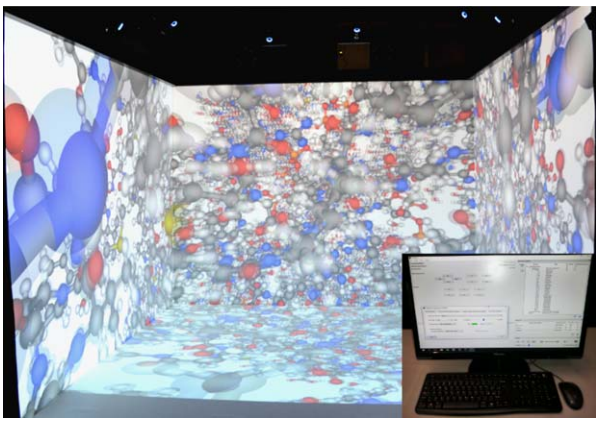

FIGURE 1 A ball-and-stick view of a macromolecular system represented in the CAVE installation at SNS with Caffeine. On the bottom right, the CAVE "control panel" is shown. The user interface of the control panel is, in large part, the same of the "desktop version" of Caffeine (see Figure 5)

for example, those related to the analysis of quantum mechanical (QM) calculations (e.g., the minimum energy path of chemical reactions) or classical simulations (such as the total energy related to a molecular dynamics [MD] trajectory).

In this article, we present the current status of Caffeine, showing its usage through the application to several test cases, which encompass many of the common levels of molecular representation and theory, including both QM studies and molecular mechanics (MM) investigations of large systems. We start from small, isolate molecules, on which high level calculations could be performed, and proceed to intricate macromolecular entities, thus involving different level of data complexity. The article is organized as follows: in a first part, some technical details of the Caffeine program and its versions for desktop and IVRs are presented; in the second part, the graphical representations of different case studies obtained with Caffeine are reported, to demonstrate the program capabilities in chemical visualization and in the analysis of results. We also make the case for potential benefits coming from the use of IVR in *setting up* simulations for medium size or large molecular systems through the examination of the dissociation mechanism of the intercalating drug doxorubicin (hereafter DOX) from a DNA fragment. General remarks and future perspectives on the development of our molecular graphics system are given in the conclusions of the article. Further technical details on the methods and algorithms employed for the visualization of molecular structures and isosurfaces are reported in the appendices.

It is important to highlight here that a rigorous user evaluation study is not our present priority. Here, we want to show the general opportunities offered by the adoption of IVR technologies in molecular modeling and, at the same time, illustrate the features of Caffeine to a wide audience.

2 | RELATED WORK

The representation of molecular structures by means of virtual reality (VR) technologies is not a new methodology: visualizations of atomistic simulations within immersive theaters were already reported at the middle of the 90s (see as an illustrative example the work done by Disz et al.[15]). Nevertheless, IVR tools did not know the diffusion that they deserved within scientific fields for at least a decade, partly due to limits of the underlying hardware and partly to the infancy of software using such technologies.[16] The growth of computer power in the last decade made possible to use IVR for rigorous scientific visualization. However, the adoption of IVR tools in molecular sciences is still an ongoing process, even if the usefulness in visualizing large systems of chemical interest (highlighting both structural and functional properties) within immersive environments has already been demonstrated.[17] Recently, Reda et al.[18] developed an application for the interactive visualization of MD simulations in ultra-resolution immersive environments, exploiting an hybrid representation which combines balls-and-sticks with volume rendering of approximate electron densities.

Among popular molecular viewers, VMD[19] supports several VR technologies[5] such as CAVE systems and ImmersaDesk,[20] using VR toolkits like FreeVR[21] and CAVElib.[22] Recently, Stone et al.[23] implemented an experimental version of VMD combining omni-directional stereoscopic visualization via head-mounted displays (Oculus Rift DK2), with ray-tracing rendering computed by a remote GPU cluster.

To confirm the interest in the use of VR environments, it is worth noticing that some commercial molecular graphics systems like Amira[24] and YASARA[25] support VR technologies. Also PyMOL[26] has a VR plug-in developed by Virtalis.[27]

3 | THE CAFFEINE MOLECULAR VIEWER

Caffeine is a new molecular viewer specifically designed and developed to take advantage of modern IVR technologies. It is implemented in C++, using the Qt framework,[28] Open Babel[29] as base cheminformatics library (we are evaluating its extension or substitution to provide a solution more suited to our needs; however Open Babel provided sufficient flexibility for the early stages of development), OpenSceneGraph[30] as 3D graphics engine, the OpenGL Mathematics library[31] and the Qt Widgets for Technical Applications (Qwt) library.[32] We use Stride[33] (as an external program invoked by our application) for detecting the secondary structures of polypeptides.

Caffeine can visualize both static and dynamic molecular structures (trajectories) read from PDB,[34] XYZ (xmol format),[35] and Gaussian Cube[36] files. Like most molecular viewer, Caffeine supports the most diffused graphical representations of molecular structures, such as "all-atoms" visualization (balls-and-sticks, licorice, and van der Waals spheres) and ribbon diagrams of polypeptides and polynucleotides. In addition, volumetric datasets such as electron densities and molecular orbitals can be imported from Gaussian Cube files and visualized as isosurfaces (several examples are presented in the Case Studies section). In the case of dynamic molecular structures, the graphic geometry is generated on the fly at each time-step, so to avoid to fill the graphic memory in the case of long trajectories. Although this is not a completely "*out-of-core*" solution (since the entire trajectory is completely loaded in main memory), it is a first step in that direction.

There are two main versions of Caffeine (sharing most of the code base and features): one for desktop computers and one designed for

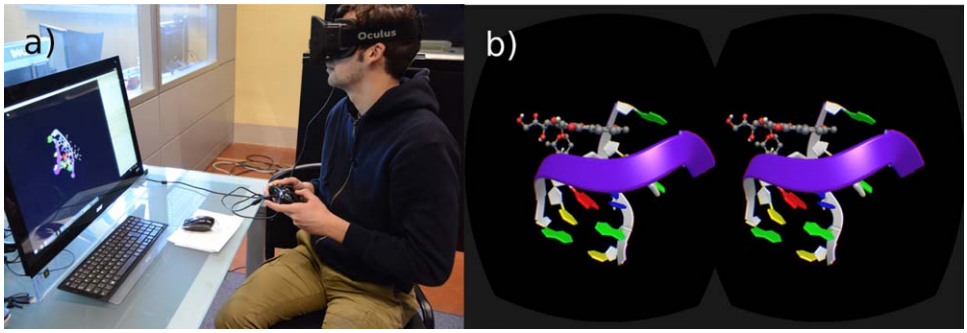

FIGURE 2 (a) User wearing the Oculus Rift DK1 helmet. (b) The DNA/DOX binary complex rendered for the Oculus Rift DK1 with an experimental version of Caffeine

VR systems equipped with multiple displays, such as the CAVE, a cubic room-sized IVR system whose walls (from three to six) are projected with stereoscopic images. Our CAVE installation has four walls/stereo projectors and it is equipped with an OptiTrack[37] tracking system (see Figure 1). The IVR version implements some functionalities which are specifically tailored for CAVE-like systems, like keystone correction of the generated images and the ability to adjust the view point and the projection parameters of the virtual camera as a function of the position and orientation of the user's head (provided by the tracking system), so to obtain a convincing and coherent stereoscopic visualization across the screens.[38] Within the CAVE the user can control the visualizer by means of a simple, "button-based," remote application for mobile devices (e.g., tablet), which allows to rotate, translate and scale the molecular system, and to control the playback of trajectories. The remaining features are accessible via a separate control window, which is displayed, in our installation, in a standard monitor placed outside the CAVE (but driven by the same workstation), as shown in Figure 1.

Note that the "CAVE version" of Caffeine can be configured to be used in systems having a different number, layout and kind (monoscopic or stereoscopic) of displays, with or without tracking systems, as long as all the monitors are driven by a single computer. In the past, our CAVE system was driven by a cluster of workstations (one for each projector) connected through a local network. This configuration required the development of distributed applications, which are notably harder to implement, debug, and extend as compared to their nondistributed counterparts. To facilitate the development of Caffeine, and to minimize the differences in the source code between the CAVE and the desktop version, we designed and developed a *distributed scene graph* library.[14] However, thanks to the latest advances in video card technology, we are now able to drive all the four projectors of our CAVE with a single computer equipped with multiple NVIDIA Quadro GPUs in scalable link interface (SLI)[39] configuration. This solution has allowed us to remove the distributed scene graph from the project, thus saving the time needed for its further improvements and extensions.

While CAVE-like systems are among the most advanced IVR systems available today, they are (very) expensive fixed installations. For that reason, they can be found only in few specialized research centers. However, thanks to the technological evolution, several companies are now developing VR headsets (primarily intended for the video game market) such as the *Oculus Rift*[9] or the *HTC/Valve "Vive."*[10] Thanks to their relatively low cost (less than a thousand US dollars) and good portability, these devices will probably gain a wide adoption in the next few years, not only in the consumer market but also for educational and research purposes. For these reasons, we are interested to support this kind of IVR helmets in Caffeine. Currently, the *Oculus Rift Development Kit 1* VR helmet is supported by an experimental desktop version of Caffeine (see Figure 2). This version is considered "experimental" since it belongs to an older development branch with respect to the current main version and it still lacks some of the latest features, such as charts and key-frames (*vide infra*). We plan to officially support some of these devices in the near future.

3.1 | Visualization of molecular structures

To obtain an interactive visualization of a large number of spheres and cylinders, resulting from the "all atoms" representation of complex molecular structures, we have developed a set of *GPU shaders* (A "*shader*" is a program running on the Graphics Processing Unit (GPU), which allows the programmer to define a specific algorithm for drawing the objects it is applied to) implementing a technique known as "*GPU-based ray-casting of quadric surfaces*." This method has been first introduced by Gumhold[40] to render a large number of ellipsoids representing symmetric tensor fields, and later generalized by Toledo and Lévy[41] so to ray-cast any quadric surface on the GPU. This technique provides very good results (both in terms of performance and image quality) when applied to molecular visualization[42–46] and, according to Kozlíková et al.[47] in a recent survey on the subject, it still represents the state of the art. A brief discussion on the implementation of this method, with specific reference to Caffeine, can be found in Appendix A.

As regards to ribbon diagrams, we implemented an algorithm for generating three-dimensional "paths," formed by a sequence of traits. Each trait is defined by two endpoints, an orientation (normal) vector, the shape of its transversal profile (circular or rectangular with rounded corners) and a set of attributes (such as sizes and colors) for each of its two ends. The traits can be rectilinear or they can be defined by cubic B-Splines, so to obtain a smoother path. In the latter case, also the normal vectors between consecutive traits are interpolated using the cubic B-Spline equation, similarly to what done by Krone et al.[49] in their algorithm for GPU-based ribbons visualization. Although our algorithm could be used to represent various kinds of information (e.g., field lines), it is currently used to generate ribbon diagrams of polypeptides

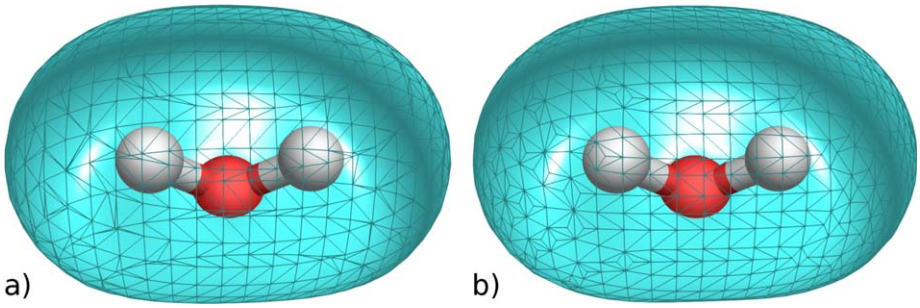

FIGURE 3 Representation of scalar volumetric datasets: comparison between the triangulation of a water molecular orbital generated by the Marching Cubes (a) and the one generated by the Surface Nets algorithm (b). Note that the traditional Marching Cubes generates many thin triangles, while Surface Nets provides a more regular tessellation. The orbitals have been computed at HF/STO-3G level of theory with Gaussian 09[56] suite of programs

and polynucleotides. For polypeptides, we followed the popular procedure described by Carson and Bugg[50]: the path is defined by the sequence of alpha-carbons in the chain, while the oxygen atoms of the backbone are used to compute the normal vectors of the peptide planes, which define the orientation of the traits. Shape, sizes, and colors of each trait are set according to the secondary structures to be presented. Path and normal vectors are finally smoothed using the B-Spline equation. In the case of polynucleotides, we used a similar procedure: phosphorus atoms replace the alpha-carbons in the definition of the path (with the exception of the first and of the last nucleotides of the chain, where the C5′ and C3′ atoms are used respectively as starting and ending points of the path), while the vector (C3′–C1′) of each nucleotide defines the normal vector of the corresponding trait.

3.2 | Isosurfaces extraction and visualization

Several molecular descriptors, such as electron density, electrostatic potential and molecular orbitals are actually examples of *volumetric datasets*. A volumetric dataset can be defined as a set of pairs $<P_i, V_i>$, where P_i is a point in the three-dimensional space and V_i is its associated value (e.g., a scalar value, a vector etc.).[51] The pair $<P_i, V_i>$ is called "voxel" (short for "volume element"). Volumetric datasets can be obtained by sampling the value of some function or measurable quantity at certain locations of the three-dimensional space. Although sampling locations could be chosen randomly, it is common to sample the data at uniformly spaced intervals, so to obtain a regular grid of voxels. It is noteworthy that, even if the values are sampled at discrete locations, it is still possible to approximate the value of a generic point lying inside a cell of the grid by interpolating the values of the eight vertices (voxels) of that cell.

At present, Caffeine visualizes scalar volumetric datasets only in the form of isosurfaces. User can chose between two extraction algoritms: the traditional *Marching Cubes*[52] and a simplified version of the *Surface Nets*.[53] With regard to the Marching Cubes we use (a slightly adapted version of) the popular implementation by Cory Gene Bloyd and Paul Bourke,[54] while for the Surface Nets we re-implemented in C++ the so-called "*Naive*" version by Mikola Lysenko[55] (originally coded in JavaScript). The Surface Nets algorithm produces a more regular triangulation than the traditional Marching Cubes method, as shown in Figure 3. Furthermore, according to some tests performed on our implementations, Naive Surface Nets results slightly faster. A brief description of these algorithms together with a first performance comparison of their implementation in Caffeine is reported in Appendix B. Further details on these and other extraction algorithms can be found in references 58 and 59.

When visualizing surfaces related to molecular properties, such as molecular orbitals, it is important to let the user perceive the relation between these surfaces and the molecular structure they refers to. For this reason surfaces are often drawn as semitransparent objects. In interactive computer graphics, semitransparent objects are usually simulated using a technique known as "*alpha blending*," first introduced by Porter and Duff[60] in 1984 and nowadays supported natively by the graphics hardware. However, a simplistic use of *alpha blending* may lead to graphical artifacts, as discussed in Appendix C. Several GPU-accelerated techniques has been proposed to properly simulate semitransparent surfaces in real-time. These techniques are generally known as "order independent transparency" (OIT) methods. Interested readers can refer to reference 66 for a comprehensive survey on the subject. We employed a method called "*Weighted Blended Order-Independent Transparency*" by McGuire and Bavoil,[64,67] because it provides a good balance between quality of the results, performance, and implementation complexity.

In Caffeine, the user can choose between two different transparency modes for isosurfaces: the traditional uniform transparency (shown in Figure 4a), where all the fragments resulting from the rasterization of the iso-surface have the same opacity (chosen by the user), or a so-called "*smart transparency*" (shown in Figure 4b), where the opacity of each fragment is a function of the dot product between the normal and the "vector to the viewer" (The vector to the viewer is the normalized direction from the position fragment to the position of the virtual camera. More precisely, both the normal and the vector to the viewer are computed by the *vertex shader* on a per-vertex basis, interpolated by the graphics card, and finally passed to the fragment shader as per-fragment values), so to highlight surface edges while clearly showing the molecular structure behind the surface. This latest method, sometimes called "*X-Ray effect*," is actually an old trick of computer graphics and it is employed also by other modern molecular viewers (such as Molekel[69] and Avogadro[70]) to represent molecular orbitals and surfaces. From our experience, "*smart*"/"*X-Ray*" transparency produces clearer, more understandable images with respect to

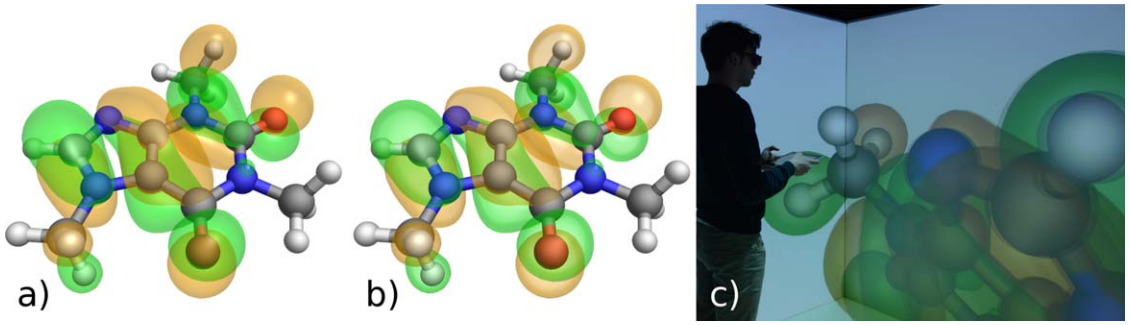

FIGURE 4 Highest Occupied Molecular Orbital (HOMO) of caffeine molecule represented as semitransparent isosurfaces (iso-value ±0.02) and drawn using the *Weighted Blended Order-Independent Transparency* method. (a) Uniform transparency: the entire surface has a constant opacity (35%). (b)"*Smart*" transparency: the regions of the surface which are orthogonal to the view direction have an lower opacity (20%), while the regions tangent to the view direction have an higher opacity (35%). (c) Same as (b) in the CAVE. Optimized molecular structure and HOMO have been calculated with PBE0 exchange and correlation functional[68] and 6-311++G(d,p) basis set with Gaussian 09[56] suite of programs

uniform transparency (see Figure 4), especially in VR systems requiring active shutter glasses (like our CAVE), where original colors are partially filtered/distorted.

3.3 | Caffeine desktop version

Figure 5a shows a first screenshot of Caffeine running as a standard desktop application, with specific hallmarks highlighted when drawing the DOX/DNA binary complex from a PDB file. Many different visualization features are available, like the possibility to modify ribbons settings and color styles.

Molecular simulations produce a lot of numerical data related to the physicochemical properties of the system taken into account. These measurements, if referred to two-dimensional scalar quantities, can be passed as input to Caffeine (using a specific file format) which plots them as line charts (as shown in Figure 5b). Furthermore, if a linear relationship exists between the measured quantity and the snapshots of a trajectory (i.e., if the quantity is a function of time), this relationship can be explicated in the dataset file. By doing so, a marker is drawn on the line chart during the playing of the trajectory, showing the value of the measured quantity in the currently displayed snapshot (see the top line chart shown in Figure 5b). This direct correlation between the currently represented structure and charted data encourage the user to exploit his "chemical intuition" and pinpoint any perceived interesting feature in the displayed system.

In addition, it is possible to "mark" a subset of the frames of the trajectory, which are deemed to be relevant for the study of the system. These "*key-frames*" may be either supplied using an additional input file or selected within Caffeine. Then, the user can (re-)visualize either the entire trajectory or only these selected key-frames. In both cases, the visualization can be performed as an animation or by explicitly skipping from one (key-)frame to the previous/following one of the sequence. Key-frames may represent particularly relevant conformations along a single trajectory but may also come from different data sources by assembling in one artificial trajectory, for example, the results of a clustering analysis over related systems. This allows for an interactive filtering of a trajectory or of any dataset since it makes it possible to associate different conformations to the same dataset and test for different hypotheses. A word of caution is needed here: while the user is free to associate key-frames to supporting data, these models must (currently) contain the same (sub-)set of atoms, and this requires a limited manipulation of the PDB files.

3.4 | Caffeine within an IVR environment

In immersive environments a new feedback, *proprioception*, is added to the perception of data. Proprioception is the capability to perceive and recognize the position of the own body in space, even without sight: the kinesthetic inputs from mechanoreceptors in muscles, tendons and joints, contribute to the human perception of limb position and limb movement in space.[72] Proprioceptive sense helps the user, without conscious efforts, to understand and evaluate the geometric properties of the visualized objects.

Within the CAVE, 2D data charts are drawn in the 3D scene in front of the user, and follow the movements of the user's head in a way similar to an augmented reality content (see Figure 6). As shown, only one chart is displayed, notwithstanding the ability of the user to switch interactively between the available charts by means of the remote application. The use of key-frames, by itself a useful feature, becomes critically important within an IVR environment since it allows the user to concentrate on the most important properties. As already stated in previous sections, the user is able to interact with the projected system through a mobile device (i.e., a tablet), so to regulate the displayed data according to the need (Figure 6b). The mobile application currently allows the user to rotate, translate and scale the molecular system, and to control the playback of frames.

As an example of the use of Caffeine with another type of IVR system, we can mention the screenshot of the DNA/DOX binary complex rendered for the *Oculus Rift DK1* already shown in Figure 2. In this case the user can interactively play the trajectory with the keyboard or, using a little trick, with a gamepad (to use the joystick an external program that maps gamepad input to keyboard events, such as JoyToKey,[73] is needed).

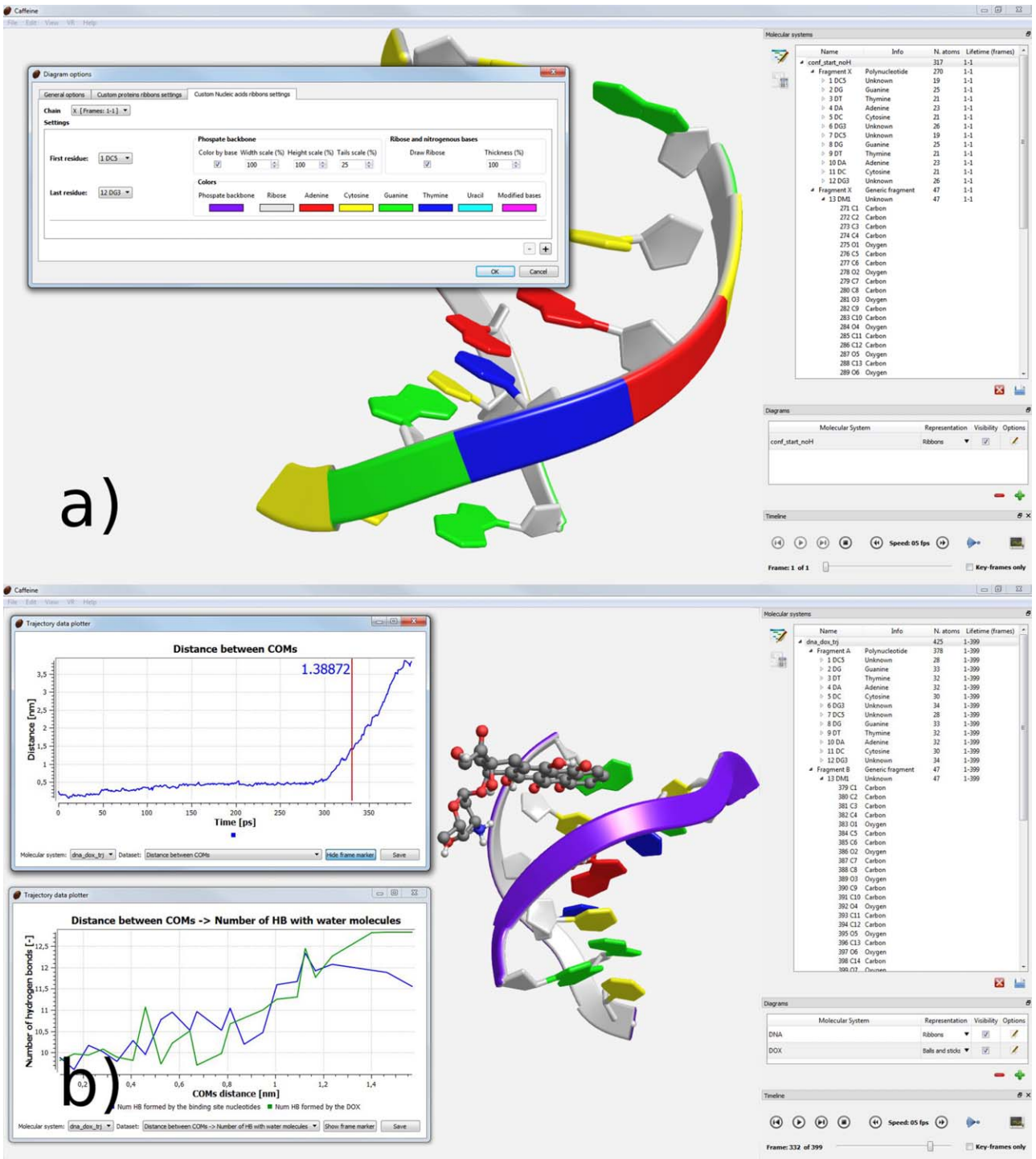

FIGURE 5 Screenshots of the current Caffeine desktop version (running in a Windows environment), used in this work for pictures realization and data analysis. The icons used in the program belongs to the Oxygen Project.[71] DNA nucleobases are colored by type. (a) Many interesting features are highlighted, such as the possibility to modify DNA ribbons appearance, according to the user needs. (b) The 2D scalar datasets associated to the molecular system are plotted as line charts. The bottom chart shows the same data of Figure 10d, while the top chart shows the distance between the centers of mass of DOX and of the DNA binding site. In the latter, the possibility to investigate the variation of the scalar quantity frame-by-frame is pointed out

3.5 | Performance evaluation

To evaluate the performance of Caffeine, we performed a benchmark to compare it with VMD.[19] The comparison is relative to the frame per seconds reached by both visualizers when rendering both static and dynamic molecular systems. It is important to note that, while obtaining a high frame rate on desktop is not an essential feature for the user, in the case of IVR systems this is critical to preserve the sense of immersion.

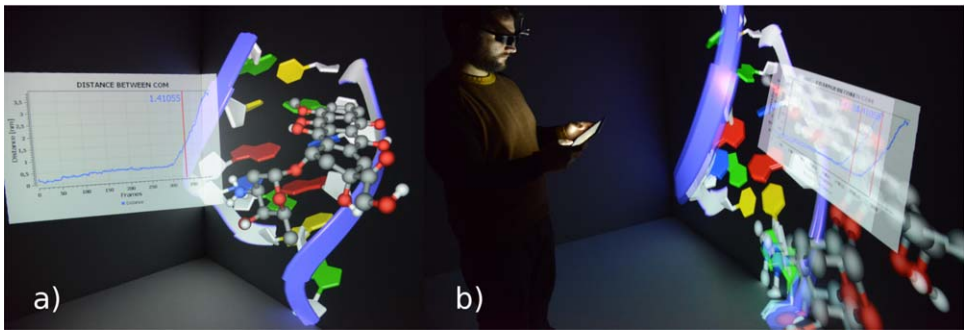

FIGURE 6 Dissociation of DOX from the DNA binding site. (a) Simultaneous representation of charted data and molecules. The binary complex is on the right while a graph showing the distance between COMs is visible on the left with the red marker highlighting the current frame and distance value. For sake of clarity, the stereo mode of projectors was temporarily disabled to shoot this photo. (b) User interacting with the DNA/DOX binary complex using a tablet, the graph is the same of the previous panel

Both applications have been configured so as to produce screen images as close as possible. In particular, this includes:

- Manually setting the "display" options and the transformation matrices of VMD to match the view and projection parameters chosen in Caffeine. Where that was not possible (i.e., the camera position of VMD is not configurable neither via GUI nor via scripting), we changed the configuration of Caffeine instead.
- Disabling "depth cueing" and axis rendering in VMD, since these features are not available in Caffeine.
- Setting the "render-mode" of VMD to "GLSL," so as to enable ray-casting of spheres and high-quality per-pixel lighting of geometry[74] (similarly to Caffeine).
- For static molecular systems, we also enabled the so-called "cache-mode" option of VMD, so as to use a display list caching mechanism to accelerate rendering of static geometry[74] (although we did not notice any variation in performance).

The computer on which the benchmark was performed is equipped with two Intel Xeon E5462 processors with a clock frequency of 2.8 GHz, 24 GB of RAM, a NVIDIA Quadro 6000 GPU, and Windows 7 Professional as operating system. Images were rendered full-screen, at a resolution of 1920 × 1080 pixels. Actually, VMD had a little advantage here, since we were not able to hide the title bar of its rendering window, so it rendered images at a slightly lower resolution.

Table 1 summarizes the frames per seconds rendered by the desktop versions of the two molecular viewers in the case of static molecular structures. To perform this benchmark we chose two medium-large assemblies from the Worldwide Protein Data Bank: (i) 1AON[75] composed by 58,870 atoms (without hydrogens) and (ii) 5AOO[76] composed by 356,280 atoms (without hydrogens). These two systems have been drawn according to the "Space filling" (van der Waals spheres), "Balls-and-Sticks" and "Ribbons" representations (and their equivalents in VMD: "VDW," "CPK," and "New Cartoon"). In VMD, we used the default quality settings for each representation, apart from the use of ray-casted spheres in place of tessellated ones. In Caffeine it is possible to specify quality settings only for ribbon diagrams, and we set them to the maximum quality. As one can see from Table 1, Caffeine is significantly faster than VMD when drawing static molecular systems.

Table 2 shows the results obtained during the visualization of a trajectory related to the dissociation of DOX from the DNA binding site (presented later as a case study). The system is composed by a fragment of DNA (378 atoms), a DOX molecule (69 atoms) and 8987 molecules of water (26,961 atoms). The trajectory contains 201 frames. For this system, we employed a mixed representation: ribbons for DNA ("New Ribbons" in VMD), balls-and-sticks for the DOX ("CPK" in VMD) and licorice for the water molecules. Furthermore we disabled the "cache-mode" option of VMD, as suggested by the user's guide[74] when dealing with trajectories. Both viewers have been configured to play a frame of the trajectory for each rendering frame, to reveal the maximum frame rate for the trajectory. In this scenario, Caffeine is slightly slower than VMD, even if the obtained frame rates have the same same order of magnitude. However, it is clear that visualization of dynamic structures in Caffeine, although fast enough in many cases, needs further optimization.

Tables 3 and 4 show the performance (min/max frames rate) of Caffeine in our CAVE system, when visualizing static and dynamic data. Overall, performances are comparable to those obtained on the

TABLE 1 Comparison of the performance between the desktop versions of Caffeine and VMD when visualizing static molecular structures

		Molecular system 1AON	5AOO
	Number of atoms	58,870	356,280
	Representation	**fps**	**fps**
Caffeine	**Space filling**	87	55
	Balls & Sticks	185	71
	Ribbons	273	128
VMD	**VDW**	17	3
	CPK	12	2
	New Cartoon	230	7

The results are expressed as frames per seconds. Different graphical representations have been considered. In VMD we chose the representations that more closely resemble those of Caffeine.

TABLE 2 Comparison of the performance between the desktop versions of Caffeine and VMD when visualizing a trajectory resulting from molecular dynamics

	DNA-DOX complex trajectory Max trajectory frames per seconds
Caffeine	12
VMD	17

The tested system is the DNA-DOX complex simulation of a single umbrella sampling window (more details in Case Studies section). The results are expressed as the maximum trajectory frames displayed in a second.

desktop system, thus allowing a comfortable and fluid immersive experience (although presenting the limits reported above for trajectories). The computer driving our CAVE is equipped with two Intel Xeon E5645 processors with a clock frequency of 2.4 GHz, 24 GB of RAM, two NVIDIA Quadro M6000 GPUs in SLI configuration, and Windows 10 Enterprise as operating system.

3.6 | Case studies

In this section, we show some useful graphic representations of simulation results obtainable with Caffeine. In the first section, we include a spin-density evaluation of 2,2,6,6-tetramethylpiperidine-1-oxyl-4-amino-4-carboxylic acid (TOAC).[77] In the second one, we used the results of a recent study on Cytochrome P450 2B4.[78] As a last case study, we present the whole DNA-DOX investigation, from the simulation setup performed within the CAVE to the final analysis.

3.7 | Spin density visualization

The analysis of isosurfaces, obtained by QM calculations or classical simulations, is particularly important in molecular sciences. In fact, the graphic representation of molecular orbitals, electron densities or electron localization functions provides a valuable help to characterize structural properties, to describe molecular interactions and to interpret spectroscopic properties. Also in the case of classical simulations the availability of volumetric datasets allows to visualize a series of useful properties such as, for example, electrostatic potential, molecular cavities (see Figure 9) or average density/occupancy (e.g., when calculating spatial distribution functions) near a selected site. As a first example of volumetric data we present the results obtained for the 2,2,6,6-tetramethylpiperidine-1-oxyl-4-amino-4-carboxylic acid (TOAC) molecule, which is characterized by

TABLE 3 Performance of Caffeine in visualizing large assemblies (PDB 5AOO) in our CAVE system; minimum and maximum frame rate for van der Waals, Balls-and-Sticks, and Ribbons representations

	5AOO	
	Min fps	Max fps
Space filling	31	73
Balls-and-Sticks	35	62
Ribbons	51	85

TABLE 4 Performance of Caffeine in visualizing a trajectory resulting from molecular dynamics (DNA-DOX complex) in our CAVE system; minimum and maximum frame rate

DNA-DOX trajectory	
Min fps	**Max fps**
10	12

the presence of a nitroxide moiety and by the possibility to be inserted into polypeptide chains substituting a natural amino acid.[77,79] It is a stable radical, which can be employed as probe in electron spin resonance measurements, allowing to obtain useful information on the conformation of the studied peptide molecule. Density functional theory (DFT) calculations have been revealed effective in the description of the electronic structure of TOAC,[77,80,81] allowing to determine that the unpaired electron occupies an anti-bonding π^* molecular orbital localized on the oxygen and nitrogen atoms of the nitroxide moiety (Figure 7a). The correct computation of the TOAC electronic structure is particularly important, because the magnetic properties, which allow to employ this molecule as spin label, are ruled by its singly occupied molecular orbital (SOMO). A similar method to visualize the unpaired electron localization is through the spin density, sketched in Figure 7b. It is interesting to note (see Figure 7c) that the spin density is essentially the same also introducing TOAC in a polypeptide chain, confirming the importance of this radical in conformation studies.

3.8 | Structure and dynamics of large systems

Enzymes belonging to the Cytochrome P450 hemoproteins family are devoted to the oxidation of a wide range of organic compounds, from drugs to environmental pollutants. Hydrophilic channels, connecting the active site to the protein surface allow the buried heme group to react with the various substrates.[82] The role of the conserved Phe429 on the catalytic activity has been thus investigated through extensive classical MD simulations and clustering analysis of the wild type (WT) together with other four different mutants, highlighting structural and hydrogen bonding observable differences.[78,83] The single point mutation was identified to be responsible of several long-range effects, including the topology of the functional aqueous accesses to the catalytic site. To get light on this, each MD frame of each of the five mutant MD simulations was analyzed to find possible tunnels regulating the access to the heme ion. In a second step such identified tunnels were connected to real pathways by means of an average linkage clustering technique. A total of four major channels were found. In particular, one major pathway was detected in all the five MD simulations, featuring different average bottleneck radius values as we proceed from the WT (lower average radius observed) to the mutants. The different behavior among the mutants was explained in terms of subtle alterations in the hydrogen bond network, that propagated along the systems and affected the whole geometries.

Since IVR environments are well suited for deep investigations on geometries of large systems, we present feasible applications of our immersive tool in the visualization of different cluster members,

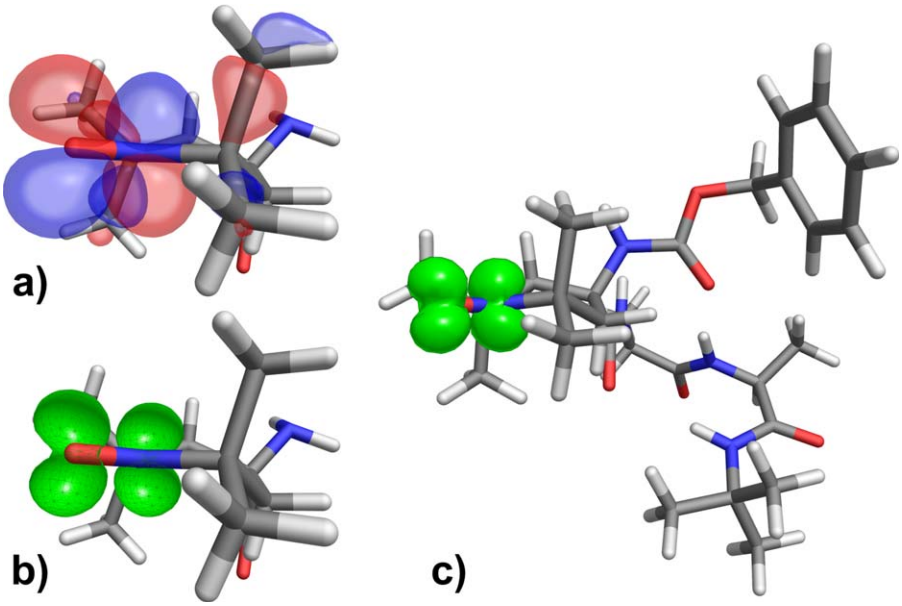

FIGURE 7 (a) and (b) represent SOMO orbital and spin density of TOAC, respectively, while (c) is the spin density of the Z-TOAC-(l-Ala)$_2$-NHtBu. Structural and electronic properties have been computed at PBE0/6-311++G(d,p) level of theory with Gaussian 09[56] suite of programs

pointing out the detection of cluster centroids and the structural differences between them (see Figure 8). Each MD frame, previously included in one specific cluster, is showed together with an index, which specifies its cluster membership. The user can switch from one cluster to another one, and select in this really feasible and nice way eligible key-frames to represent cluster centroids or outlying "extreme" conformers in clusters, to describe the differences within the obtained clusters and the relationships between the various mutations. The detected channels can also be visualized in Caffeine thanks to the support for Gaussian Cube file format.[36] To be supported in Caffeine, the channels identified in the previous study have been thus converted in Guassian Cube format, thanks to an *in-house* script, which computes density values using a simple Gaussian function taken from the literature.[84] As shown in Figure 9b, the difference and peculiarities among the mutants can be highlighted by displaying simultaneously the relative structures within the CAVE. Such investigations could be easily performed thanks to IVR technologies: on the contrary, the complex structures of this dataset could lead to unpleasant misunderstanding using a 2D computer desktop, mostly when the same analysis is explained to nonexperts.

3.9 | Dissociation of DOX from the DNA binding site

Intercalating drugs act as inhibitors of Topoisomerase I or II (or both). DOX is an antraciclynic intercalating drug whose structure can be divided in (i) a planar hydrophobic part, constituted by an antraquinone ring system, and (ii) a hydrophilic aminosugar moiety. Intercalating drugs bearing fused (hydrophobic) ring systems can insert between base pairs through the creation of favorable π-stacking with nearby nucleobases.[85] We report in this section, a study about the unbinding process of DOX from DNA, to show the effectiveness of IVR tools within a computational research project. The unbinding process of DOX from the binary complex was here investigated with umbrella sampling,[86] using the distance between the center of mass (COM) of

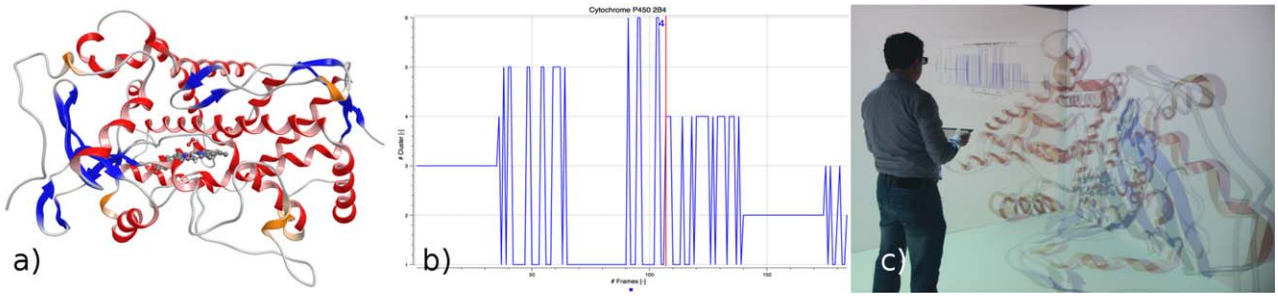

FIGURE 8 (a) Cytochrome P450 2B4 WT structure shown using ribbons. The heme group and OOH^- anion are shown in balls-and-sticks. (b) Distribution of conformations in clusters along the artificial trajectory created by sampling structures from the original WT and the four mutants simulations. Less frames were used with Caffeine, obtained from a uniform re-sample of the original clusters. (c) Comparison of the differences between the obtained clusters using virtual reality with Caffeine

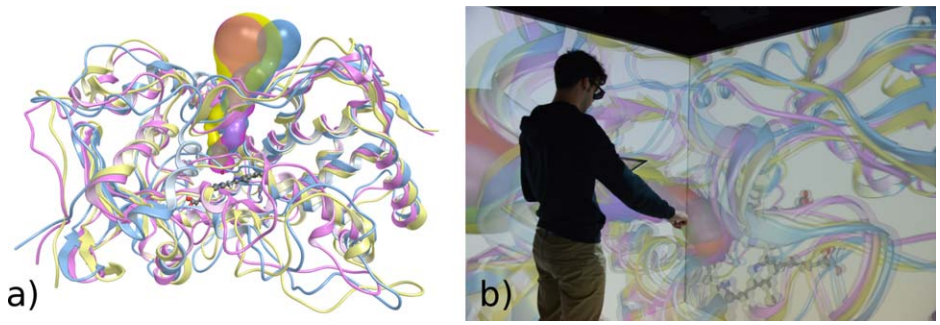

FIGURE 9 Superposition of representative structures, together with relative water channels, of three different mutants of Cytochrome P450 2B4, both on a standard desktop (a) and within the CAVE (b)

DOX and of the binding site as the reaction coordinate. The selection of the starting configurations for the umbrella windows was performed with our IVR environment: the ability to view, at the same time, a molecular conformation and the chart reporting the distance between COMs was exploited to select sensible structures in a very accurate way (see Figure 6).

These structures were marked as key-frames and used for the subsequent umbrella simulations (details about the simulations are

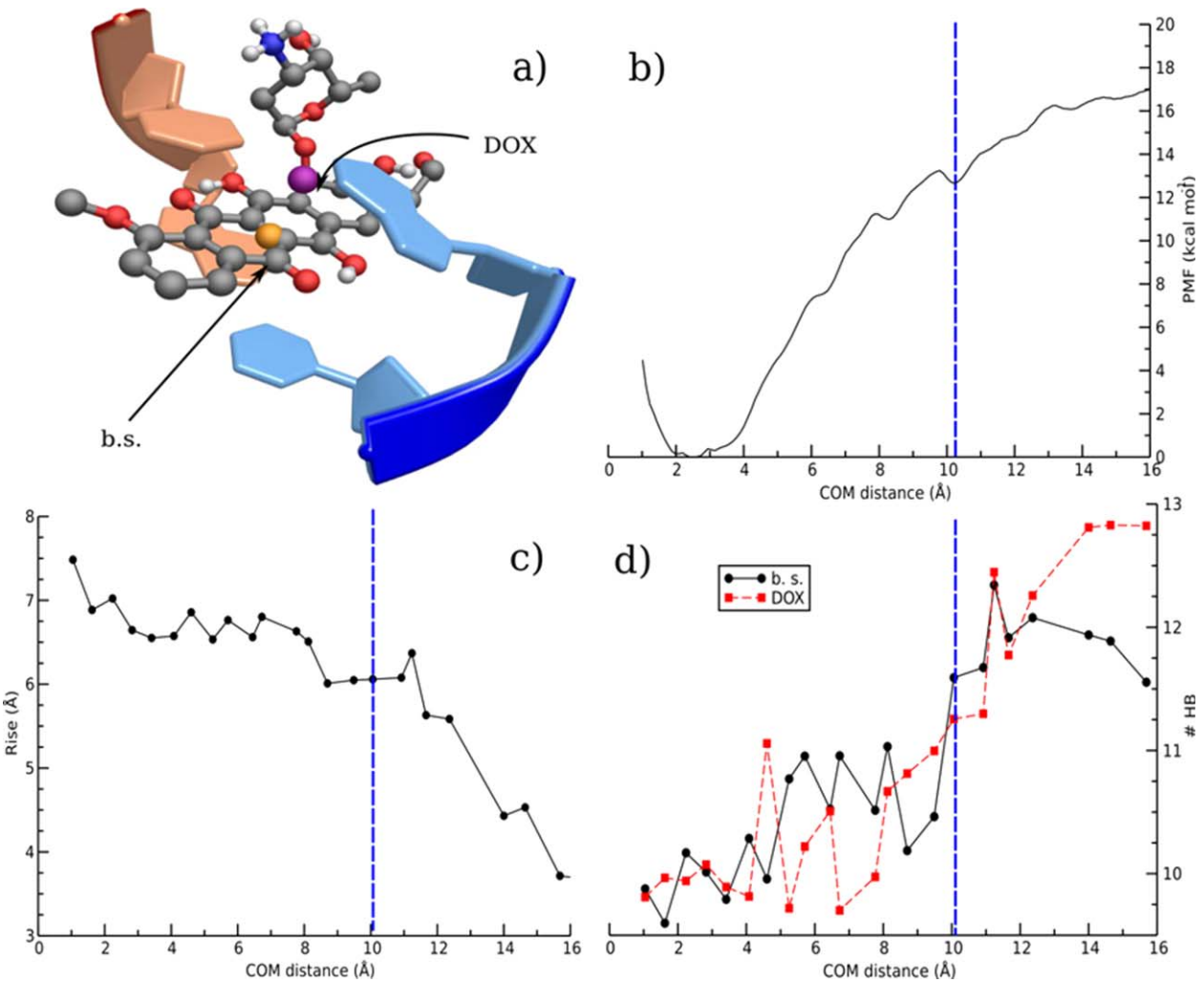

FIGURE 10 Binary complex dissociation process. Note that in panels (b-d) the position of the intermediate at 10 Å and final dissociated state along the sampling coordinate is highlighted with a blue dashed line. (a) Binding site of the DOX compound in the initial conformation. The position of the binding site COM (b.s., gold sphere) and of DOX (purple sphere) is shown. (b) Potential of mean force (PMF) curve associated to the distance between centers of mass of the binding base pairs and of the DOX drug. (c) Calculated change in rise between DNA base pairs. (d) Number of hydrogen bonds between the binding site nucleobases (black) or DOX (red) with water molecules

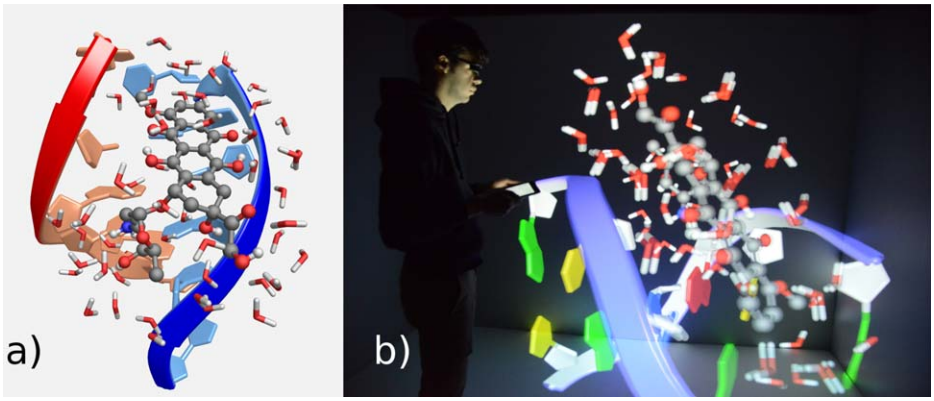

FIGURE 11 (a) Intermediate bound state of DOX with the nearest neighbor water molecules shown as licorice. (b) Same as (a), within the CAVE

reported in the Supporting Information). The obtained free energy ΔG is represented in Figure 10b. Interestingly, a partially stable state was found at 10.2 Å: here the rigid body of the DOX molecule lies on the plane defined by the two DNA backbones, while the intercalation site is still in an opened conformation. This state may be associated to the intermediate one (IM) already found in the case of daunomycin.[87] Roughly, 14 kcal/mol are necessary for DOX to reach the solvent.

Since intercalation, and, subsequently, dissociation of anthracyclines from DNA has been demonstrated to alter the DNA structure,[88] we selected few parameters to measure the structural modifications that take place during the binding process, which could be used in further analysis. Figure 10c shows the average rise distance between base pairs in the intercalation site as a function of the reaction coordinate in the simulated windows. High rise values (of approximately 7.5 Å) are detected for the intercalated state: then, as DOX approaches the bulk solvent, the distance between two consecutive bases decreases, reaching a final value of about 3.5 Å in the unbound state, very close the value of 3.4 Å featured by native B-DNA. The intermediate state (whose structure could be considered as the mean structure of the umbrella window starting at 10 Å of COMs distance) is showed both on a standard desktop and using IVR in Figure 11: it is possible to observe that such conformation seems to be still in an opened conformation (assuming a rise value close to 6 Å), thus being accessible by water molecules.

The hydration of the binary system as a function of the reaction coordinate was taken into account (and plotted in Figure 10d): such property was calculated as the average number of hydrogen bonds between water molecules and either DOX or the four nucleobases that delimit the binding site. The intercalation induces a decrease of the H-bonds since DOX acts as a barrier for water molecules, which cannot enter into the binding site. At the same time, DOX is less hydrated as it approaches the DNA binding task: on average, two hydrogen bonds that take place in the DOX unbound state are not preserved in the intercalated one. It is interesting to observe a peak of average number of hydrogen bonds in proximity of the intermediate state: in fact, at this point, DOX has already left the intercalation site, so as to be considered solvent-exposed whereas the binding site is still opened. Finally, after 10 Å of COM distance, the number of water molecules H-bonded to the binding site bases slowly decreases. This is in agreement with our previous considerations: in fact, after this point, as shown in Figure 10c, the binding site reduces its size, because of the departure of DOX, thus decreasing water accessibility to the intercalation site nucleobases.

Suitably chosen conformations (i.e., the centroids of the single umbrella windows) were used to reconstruct the whole unbinding process, from the intercalated to the completely unbound state, through the IM one, so as to build an artificial trajectory to be used in Caffeine to follow in the meanwhile both chemical structure evolution and related structural parameters. Moreover, considering the PMF chart in Figure 10b, it is always possible to connect the current, visualized snapshot to its associated free energy value just switching from the COMs distance chart to the PMF one, thus increasing the user's understanding of the overall free energy study. It is important to highlight here that a quantitative evaluation of binding/unbinding of DOX was not the ambition of the present study: here we applied a simplified (to limit the computational cost), yet consistent, computational protocol for illustrating the features of Caffeine to a wide audience. It is anyway remarkable that the obtained results are in line with currently available literature data.

4 | CONCLUSION AND OUTLOOKS

In this contribution, the technical details and main features of Caffeine, a novel molecular viewer suited for IVR environments, are presented. Caffeine allows a smooth transition from desktop computers, where the most diffused molecular graphics software work, to IVR systems, such as helmets and the CAVE theater. In our opinion, it is reasonable to expect benefits from systematic use of IVR environments within computational investigations. In fact, molecules can be perceived as three-dimensional objects with a well-established position of atoms in space, thus being characterized in a more precise and effective manner. Moreover, some peculiar features of Caffeine, such as the augmented reality-like visualization of 2D charted data and the interactive filtering of trajectories with "key-frames," envision our idea of possible,

productive and realistic employment of IVR in computational chemistry, which could be seen as reliable front-end tool in post-processing analysis.

Caffeine is under active development and there are a number of new features we would like to add in the future releases. It is worth noticing that, although the current version of Caffeine allows to visualize orbitals, spectra, and time evolution of molecular properties, which can be useful in the interpretation of some spectroscopic observable, further improvements are needed for obtaining a full user-friendly virtual spectrometer.

Although, it is sufficiently fluid, the visualization of multiple structures (trajectories) would benefit from further optimization. Such an optimization can be implemented, for example, using some GPU-accelerated methods, like those proposed by Krone et al.[49] and by Wahle and Birmanns[89] for ribbon generation.

Another line of development concerns the interaction with visualized data. Right now, within the CAVE the user is able to interact with the system by means of a simple application for tablet computers. Furthermore, at present, many features of Caffeine cannot be controlled via tablet, and require the help of a second user acting on the external control panel. We want to better exploit our tracking system by enabling hands tracking within the CAVE, thereby allowing the user to manipulate the visualized system, that is, to move, scale, rotate the system and playing back or forward across frames. As for IVR helmets, we plan to officially support this kind of devices in the near future.

From a general perspective, we envision Caffeine as an advanced graphics front-end focused on visualization and interactive data handling, able to communicate with other analysis environments using flexible interchange formats. As an example, we are working on a flexible representation of hydrogen bonds from MD.[90] Other efforts are being directed in more extended visualization of volumetric and spectroscopic quantities, acting as an IVR front-end for other analysis environments.[91]

ACKNOWLEDGMENTS

The authors thank the DREAMS Lab technical staff for managing the computing facilities at SNS and Dr. Balasubramanian Chandramouli for useful discussion.

REFERENCES

[1] M. Valle, *Int. J. Quantum Chem.* **2013**, *113*, 2040.

[2] N. Luehr, A. G. B. Jin, T. J. Martínez, *J. Chem. Theory Comput.* **2015**, *11*, 4536. PMID: 26574246.

[3] C. Casher, C. Leach, C. S. Page, H. S. Rzepa, *J. Mol. Struct. (Theochem)* **1996**, *368*, 49.

[4] J. D. Hirst, D. R. Glowacki, M. Baaden, *Faraday Discuss.* **2014**, *169*, 9.

[5] J. E. Stone, A. Kohlmeyer, K. L. Vandivort, K. Schulten, In *Adv. Visual Comput.: 6th International Symposium, ISVC 2010* (Eds: G. Bebis, R. Boyle, B. Parvin, D. Koracin, R. Chung, R. Hammound, M. Hussain, T. Kar-Han, R. Crawfis, D. Thalmann, D. Kao, L. Avila), Springer Berlin Heidelberg, Las Vegas, NV November 29–December 1 **2010**, pp. 382–393.

[6] S. Rajeev, Z. Michael, V. I. Pavlovic, T. S. Huang, Z. Lo, S. Chu, Y. Zhao, J. C. Phillips, and K. Schulten, *IEEECGA* **2000**, *20*, 29.

[7] Microsoft, Kinect for Xbox One, http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one, Accessed 5 July, 2016.

[8] Leap Motion Inc., Leap Motion, https://www.leapmotion.com, Accessed 5 July, 2016.

[9] Oculus VR LLC, Oculus Rift, https://www.oculus.com/en-us/rift/, Accessed 5 July, 2016.

[10] HTC and Valve, Vive, http://www.htcvive.com, Accessed 5 July, 2016.

[11] Novint Technologies Inc., Falcon 3D Touch controller, http://www.novint.com/index.php/novintfalcon, Accessed 5 July, 2016.

[12] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, J. C. Hart, *Commun. ACM* **1992**, *35*, 64.

[13] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, in *Proc. 20th Annu. Conf. Comput. Graph. Interactive Tech.*, SIGGRAPH '93, ACM, New York, NY **1993**, pp. 135–142.

[14] A. Salvadori, A. Brogni, G. Mancini, V. Barone, in *Augmented Virtual Reality: First Int. Conf., AVR 2014* (Eds: T. L. De Paolis, A. Mongelli), Springer International Publishing, Lecce, Italy September 17–20, **2014**, Revised Selected Papers, pp. 333–350.

[15] Terrence, D. Michael, P. Rick, S. Michael, P. Valerie, T. Virtual, *Reality Visualization of Parallel Molecular Dynamics Simulation*, Society for Computer Simulation, **1995**, pp. 483–487.

[16] A. van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, R. M. Simpson, *IEEE Comput. Graph. Appl.* **2000**, *20*, 26.

[17] E. Moritz, J. Meyer, in *Proc. Fourth IEEE Symp. Bioinformatics Bioeng.*, **2004**., pp. 503–507.

[18] K. Reda, A. Knoll, K. I. Nomura, M. E. Papka, A. E. Johnson, J. Leigh, in IEEE Symp. Large-Scale Data Anal. Visualization (LDAV), **2013**, pp. 59–65.

[19] W. Humphrey, A. Dalke, K. Schulten, *J. Mol. Graph.* **1996**, *14*, 33.

[20] M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, G. L. Dawe, M. D. Brown, *SIGGRAPH Comput. Graph.* **1997**, *31*, 46.

[21] W. R. Sherman, *Daniel Coming, and Simon Su*. Freevr: Honoring the Past, Looking to the Future. Proc. SPIE, 8649:864906–864906–15, **2013**.

[22] Mechdyne Corporation, Cavelib: The ultimate solution for 3d virtual reality displays, http://www.mechdyne.com/software.aspx?name=CAVELib, Accessed 5 July, 2016.

[23] J. E. Stone, W. R. Sherman, K. Schulten, in *IEEE Int. Parallel Distributed Processing Symposium Workshop (IPDPSW)*, in Press. http://www.ks.uiuc.edu/Publications/Papers/paper.cgi?tbcode=STON2016A

[24] FEI, Amira 3D Software for Life Sciences, http://www.fei.com/software/amira-3d-for-life-sciences/, Accessed 5 July, 2016.

[25] YASARA Biosciences, YASARA—Yet Another Scientific Artificial Reality Application, http://www.yasara.org/, Accessed 5 July, 2016.

[26] Schrödinger, LLC, The PyMOL Molecular Graphics System, Version 1.8, http://www.pymol.org, Accessed 5 July, 2016.

[27] Virtalis Inc., VR For PyMOL, http://www.virtalis.com/vr-for-pymol/, Accessed 5 July, 2016.

[28] The Qt Company, Qt framework, http://www.qt.io, Accessed 5 July, 2016.

[29] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, G. R. Hutchison, *J. Cheminform.* **2011**, *3*, 1.

[30] OpenSceneGraph, http://www.openscenegraph.org, Accessed 5 July, 2016.

[31] G-Truc Creation, OpenGL Mathematics, http://glm.g-truc.net, Accessed 5 July, 2016.

[32] U. Rathmann, J. Wilgen, Qwt - Qt Widgets for Technical Applications, http://qwt.sourceforge.net, Accessed 5 July, 2016.

[33] D. Frishman, P. Argos, *Proteins* **1995**, *23*, 566.

[34] Worldwide Protein Data Bank, PDB file format, http://www.wwpdb.org/documentation/file-format, Accessed 5 July, 2016.

[35] OpenBabel, XYZ file format, http://openbabel.sourceforge.net/wiki/XYZ, Accessed 5 July, 2016.

[36] Gaussian Inc., The cubegen utility, http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm, Accessed 5 July, 2016.

[37] NaturalPoint Inc., OptiTrack - Motion Capture Systems, https://www.optitrack.com/, Accessed 5 July, 2016.

[38] R. Kooima, Generalized perspective projection, http://csc.lsu.edu/~kooima/articles/genperspective/index.html, 2008, Accessed 5 July, 2016.

[39] NVIDIA Corporation, Quadro SLI Technology, http://www.nvidia.com/object/quadro-sli-technology.html, Accessed 5 July, 2016.

[40] S. Gumhold, in *Proc. Vision, Modeling, Visualization Conf. 2003 (VMV 2003)* (Ed: T. Ertl), Aka GmbH, München, Germany November 19-21, 2003, pp. 245–252.

[41] R. Toledo, B. Lévy, Extending the graphic pipeline with new gpu-accelerated primitives. Technical report, INRIA-ALICE, **2004**.

[42] G. Reina, T. Ertl, in *Proc. Seventh Joint Eurographics/IEEE VGTC Conf. Visualization*, EUROVIS'05, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland **2005**, pp. 177–182.

[43] S. Christian, W. Tim, B. Mario, G. Markus, G, in *Proc. 3rd Eurographics/IEEE VGTC Conf. Point-Based Graphics*, SPBG'06, pages 59–65, Aire-la-Ville, Switzerland, Switzerland, **2006**. Eurographics Association.

[44] M. Tarini, P. Cignoni, C. Montani, *IEEE Transactions on Visualization and Computer Graphics* **2006**, *12*, 1237.

[45] M. Chavent, A. Vanel, A. Tek, B. Levy, S. Robert, B. Raffin, M. Baaden, *Journal of Computational Chemistry* **2011**, *32*, 2924.

[46] P. D. Bagur, N. Shivashankar, V. Natarajan, Improved quadric surface impostors for large bio-molecular visualization. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '12, pages 33:1–33:8, New York, NY, USA, **2012**. ACM.

[47] K. Barbora, K. Michael, L. Norbert, F. Martin, B. Marc, B. Daniel, V. Ivan, P. Julius, H. Hans-Christian, in *Eurographics Conference on Visualization (EuroVis) – STARs* (Eds: R. Borgo, F. Ganovelli, I. Viola), The Eurographics Association, **2015**.

[48] J. F. Blinn, in *Proc. 4th Annu. Conf. Comput. Graph. Interactive Tech.*, SIGGRAPH '77, ACM, New York, NY **1977**, pp. 192–198.

[49] K. Michael, B. Katrin, E. Thomas, in *Theory and Practice of Computer Graphics* (Eds: I. S. Lim, W. Tang), The Eurographics Association, **2008**.

[50] M. Carson, C. E. Bugg, *J. Mol. Graph.* **1986**, *4*, 121.

[51] A. Kaufman, K. Mueller. in *Visualization Handbook* (Eds: C. D. HansenChris, R. Johnson), Elsevier Butterworth-Heinemann, **2005**, pp. 127–174.

[52] W. E. Lorensen, H. E. Cline, in *Proc. 14th Annu. Conf. Comput. Graph. Interactive Tech.*, SIGGRAPH '87, ACM, New York, NY **1987**, pp. 163–169.

[53] S. F. F. Gibson, in *Proc. First Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, MICCAI '98, Springer-Verlag, London, UK **1998**, pp. 888–898.

[54] P. Bourke, Polygonising a scalar field (marching cubes), http://paulbourke.net/geometry/polygonise/, Accessed 5 July, 2016.

[55] L. Mikola, Smooth voxel terrain (part 2), http://0fps.net/2012/07/12/smooth-voxel-terrain-part-2/, Accessed 5 July, 2016.

[56] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, D. J. Fox, *Gaussian ~09 Revision D.01*, Gaussian Inc., Wallingford, CT **2009**.

[57] J. H. Han, *3D Graphics for Game Programming*, Chapman and Hall/CRC Press, **2011**.

[58] C. D. Hansen, C. R. Johnson, in *The Visualization Handbook*, Elsevier Butterworth-Heinemann, Burlington, MA, **2005**.

[59] W. Rephael, *Isosurfaces: Geometry, Topology, and Algorithms*, CRC Press, **2013**.

[60] T. Porter, T. Duff, in *Proc. 11th Annu. Conf. Comput. Graph. Interactive Tech.*, SIGGRAPH '84, ACM, New York, NY **1984**, pp. 253–259.

[61] E. Cass, *Technical Report, NVIDIA Corporation* **2001**, *2*, 7.

[62] L. Bavoil, K. Myers, Order independent transparency with dual depth peeling. *Technical Report., NVIDIA Corporation*, pp. 1–12.

[63] B. Liu, L. Y. Wei, Y. Q. Xu, E. Wu, presented at *11th IEEE Int. Conf. Comput. Aided Des. Comput. Graph., CAD/Graphics '09*, 2009, pp. 452–456.

[64] M. McGuire, L. Bavoil, *J. Comput. Graph. Tech.* **2013**, *2*, 122.

[65] M. Houman, Sort-independent alpha blending. *Perpetual Entertainment, GDC Talk*, **2007**.

[66] M. Maule, J. L. D. Comba, R. P. Torchelsen, R. Bastos, *Comput. Graph.* **2011**, *35*, 1023.

[67] M. Morgan, Casual effects: Implementing weighted, blended order-independent transparency, http://casual-effects.blogspot.it/2015/03/implemented-weighted-blended-order.html, Accessed 5 July, 2016.

[68] C. Adamo, V. Barone, *J. Chem. Phys.* **1999**, *110*, 6158.

[69] Ugo Varetto, Molekel 5.4, http://ugovaretto.github.io/molekel/, Accessed 5 July, 2016.

[70] M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, G. R. Hutchison, *J. Cheminform.* **2012**, *4*, 1.

[71] KDE, Oxygen project, https://techbase.kde.org/Projects/Oxygen, Accessed 5 July, 2016.

[72] J. L. Taylor, in *Encyclopedia of Neuroscience* (Ed: Larry R. Squire), Academic Press, **2009**, pp. 1143–1149.

[73] JTKSOFT, JoyToKey, http://joytokey.net/en/, Accessed 5 July, 2016.

[74] Vmd user's guide, version 1.9.2, http://www.ks.uiuc.edu/Research/vmd/current/ug/, Accessed 5 July, 2016.

[75] Z. Xu, A. L. Horwich, P. B. Sigler, *Nature* **1997**, *388*, 741.

[76] C. Sabin, P. Plevka, *Acta Crystallogr. Sect. F* **2016**, *72*, 188.

[77] M. D'Amore, R. Improta, V. Barone, *J. Phys. Chem. A* **2003**, *107*, 6264.

[78] G. Mancini, C. Zazza, *PLoS One* **2015**, *10*, 1.

[79] C. Toniolo, M. Crisma, F. Formaggio, C. Peggion, *Biopolymers (Pept. Sci.)* **2001**, *60*, 396.

[80] R. Improta, V. Barone, *Chem. Rev.* **2004**, *104*, 1231.

[81] S. Carlotto, P. Cimino, M. Zerbetto, L. Franco, C. Corvaja, M. Crisma, F. Formaggio, C. Toniolo, A. Polimeno, V. Barone, *J. Am. Chem. Soc.* **2007**, *129*, 11248.

[82] V. Cojocaru, P. J. Winn, R. C. Wade, *Biochim. Biophys. Acta* **2007**, *1770*, 390.

[83] D. Usharani, C. Zazza, W. Lai, M. Chourasia, L. Waskell, S. Shaik, *J. Am. Chem. Soc.* **2012**, *134*, 4053. PMID: 22356576.

[84] R. A. Laskowski, *J. Mol. Graph.* **1995**, *13*, 323.

[85] W. A. Denny, B. C. Baguley, *Curr. Top. Med. Chem.* **2003**, *3*, 339.

[86] G. M. Torrie, J. P. Valleau, *J. Comput. Phys.* **1977**, *23*, 187.

[87] M. Wilhelm, A. Mukherjee, B. Bouvier, K. Zakrzewska, J. T. Hynes, R. Lavery, *J. Am. Chem. Soc.* **2012**, *134*, 8588. PMID: 22548344.

[88] F. Yang, S. S. Teves, C. J. Kemp, S. Henikoff, *Biochim. Biophys. Acta* **2014**, *1845*, 84.

[89] M. Wahle, S. Birmanns, *Proc. SPIE* **2011**, *7868*, 786805.

[90] M. Pagliai, G. Cardini, R. Righini, V. Schettino, *J. Chem. Phys.* **2003**, *119*, 6655.

[91] D. Licari, A. Baiardi, M. Biczysko, F. Egidi, C. Latouche, V. Barone, *J. Comput. Chem.* **2015**, *36*, 321.

AUTHORS' BIOGRAPHIES

Andrea Salvadori is a PhD student in Chemistry at Scuola Normale Superiore (Pisa, Italy) in the SMART@SNS Laboratory headed by Prof. Vincenzo Barone. His primary research interests are related to the application of computer graphics and virtual reality technologies to the field of Molecular Graphics. He received his Master's Degree in Computer Science ("Laurea Specialistica in Tecnologie Informatiche") from University of Pisa (Pisa, Italy).

Gianluca Del Frate is currently a PhD student at Scuola Normale Superiore, Pisa, in the SMART@SNS Laboratory headed by Prof. Vincenzo Barone. He received his Master Degree in Medicinal Chemistry in 2013 from the University of Pisa. His research interests are related to the computational study of biochemical systems and to the development and validation of accurate force fields for molecular simulations.

Marco Pagliai obtained his PhD in Chemistry (2004) at the University of Florence (Italy), where he did postdoctoral research until 2015. He then joined the group of Prof. Vincenzo Barone at Scuola Normale Superiore in Pisa as a research fellow. His research interests are the application of classical and ab initio molecular dynamics simulations to characterize structural, dynamic and spectroscopic properties of complex systems in condensed phases.

Giordano Mancini obtained his PhD in Physical Chemistry in 2008 from the University of Rome La Sapienza. Since 2013, he has been a post doc researcher at Scuola Normale Superiore in Pisa. He works on the development of accurate force fields for classical molecular dynamics and on the application of IVR technologies to molecular modeling.

Vincenzo Barone is Full Professor in Physical Chemistry since 1994 and has been appointed as Director of the Scuola Normale Superiore in 2016. He is author of more than 700 papers with more than 40,000 citations, an h-factor of 78, and 10 papers with more than 1000 citations each. He has contributed to the development of Density Functional Theory, solvation theory, computational spectroscopy and performed state-of-the-art applications in several fields including cultural heritage and astrochemistry.

SUPPORTING INFORMATION

Additional Supporting Information can be found in the online version of this article at the publisher's website.

How to cite this article: A. Salvadori, G. Del Frate, M. Pagliai, G. Mancini, V. Barone. Int. J. Quantum Chem. 2016, *116*: 1731–1746. DOI: 10.1002/qua.25207

APPENDIX A: NOTES ON THE IMPLEMENTATION OF GPU-BASED RAY-CASTING OF QUADRIC SURFACES IN CAFFEINE

GPU-based ray-casting of quadric surfaces is a technique widely used in scientific visualization which allows to visualize a large amount of glyphs at interactive frame rates by exploiting the huge computing power of the modern GPUs. The use of quadric surfaces as glyphs is motivated by the fact that the intersection between a ray and a surface of this type can be computed by solving a simple quadratic equation.

The key idea of this method is to feed the graphics pipeline with a simple proxy geometry (e.g., a bounding box or a point sprite) in place of the desired implicit surface. The proxy geometry must be sized so to completely enclose the surface in window space, and a ray-casting shader must be enabled when drawing the geometry, so to process each fragment resulting from its rasterization. In particular, the fragment shader analytically computes the intersection between the surface and a ray starting from the virtual camera and passing through the center of the fragment. If there is no intersection then the fragment is discarded, otherwise its depth is adjusted with the one of the intersection point closest to the camera. Usually, the shader also computes the normal vector of the surface at the intersection point, that will be used to calculate the color of the fragment according to a certain shading

model (e.g., the Blinn–Phong[48] model). Although it is possible to write a single shader capable of drawing any quadric surface,[41,43,45] we chose to implement separate dedicated shaders for spheres and capsules (cylinders with (or without) semispheres at their ends). Furthermore, we wrote a geometry shader to generate the proxy geometry on the fly, similarly to what done by Bagur et al.[46] By doing so, we are able to represent a sphere with only seven floating point numbers (three for the center, three for the color, and one for its radius), and a capsule with ten floats (two vertex, color, and radius). The geometry shader will then generate a quad for each sphere and up to two quads for each cylinder as proxy geometries.

APPENDIX B: NOTES ON THE IMPLEMENTATION OF THE MARCHING CUBES AND THE SURFACE NETS ALGORITHMS IN CAFFEINE

Caffeine implements, and provides to the user, two different algorithms for the construction of a triangle mesh approximating an isosurface of a volumetric dataset: the traditional *Marching Cubes*[52] and a simplified version of the *Surface Nets*.[53] With regard to the Marching Cubes we use (a slightly adapted version of) the popular implementation by Cory Gene Bloyd and Paul Bourke,[54] while for the Surface Nets we re-implemented in C++ the so-called "*Naive*" version by Mikola Lysenko [55] (originally coded in JavaScript).

In brief, the Marching Cubes algorithm iterates over the cells bordered by the volume grid (called "cubes" even if they can be non-cubical parallelepipeds) and, for each cell, it "marks" the voxels whose value is lower than the *isovalue*. If two voxels connected by an edge have a different marking (because one value is lower than the isovalue, while the other is greater than or equal to it), then the isosurface crosses the edge. In that case, and in function of which edges are crossed, one or more triangles are generated by applying a predefined triangulation scheme. The vertices of these triangles always lies on the intersected edges and their coordinates are computed by linearly interpolating the related voxels in function of their value and of the isovalue. At the end, the algorithm returns the set of the triangles resulting from processing the entire grid, which constitutes a good approximation of the isosurface. The Surface Nets algorithm operates similarly to the Marching Cubes, by iterating over the cells and checking which ones are crossed by the isosurface. The differences between the two approaches resides in the fact that Surface Nets generates only one vertex for each crossed cell, that this vertex lies within the cell (instead to be constrained on an edge) and that the resulting sets of polygons (quads, which can anyway be splitted in triangles) are obtained by connecting each vertex with the vertices of the neighbors cells (sharing a face) crossed by the same isosurface. As regard to the position of the vertex within the cell, the original algorithm[53] initially places it at the center of the cell. Then, an iterative process is applied, which moves the vertices so as to minimize the sum of the squared lengths of the links connecting the vertices, with the constrain to keep each vertex within its original cell. The simplified "*Naive*" version, instead, choses as vertex of the cell the centroid of the approximated intersection points between the isosurface and the edges of the cells (To the best of our knowledge, Mikola Lysenko[55] was the first to propose to place the vertex at the centroid of the intersection points) (computed as in the Marching Cubes algorithm). By avoiding the iterative minimization process, the "*Naive*" Surface Nets method is both faster and easier to implement, although it could produce suboptimal results.

The Surface Nets algorithm produces a more regular triangulation than the traditional Marching Cubes method, as shown in Figure 3. Furthermore, according to some tests performed on our implementations, Naive Surface Nets results slightly faster. Let's consider, as an example, the isosurface representing the molecular orbital of a caffeine molecule with isovalue 0.02, drawn in orange in Figure 4. The traditional Marching Cubes algorithm takes 66 ms to create the corresponding triangle mesh composed by 13,896 triangles and 41,688 vertices, while the Naive Surface Nets method produces a similar isosurface composed by 13,928 triangles and 6980 vertices in 52 ms (These tests has been performed on the same desktop computer described in section "Performance Evaluation"). Note that, while the number of triangles produced by the two algorithms is almost the same, the traditional Marching Cubes method outputs much more vertices than its Naive Surface Nets counterpart. This is due to the fact that the traditional Marching Cubes algorithm does not take into account that the vertices it generates in each cell are shared among multiple triangles, hence the same vertex is repeated multiple times in the output list. In the case of the Surface Nets method, instead, it is easier to obtain an *indexed triangle list*, that is, the output vertex list does only contains non-duplicated vertices and the triangles are defined by an additional index list storing three indices for the vertex list for each triangle to be generated. *Indexed triangle list* allows to reduce the memory required to store a given geometry, and also brings an increase of the rendering performance thanks to a better exploitation of the GPU cache.[57] Regarding to generation times, the difference is due to the fact that the Marching Cubes algorithm computes per-vertex attributes (and in particular the normal vector) for each intersection point between the edges and the isosurface (consequently, several times for the same geometrical vertex within the cell), while in Surface Nets method these attributes are computed one time only.

APPENDIX C: SIMULATING SEMITRANSPARENT SURFACES IN REAL-TIME 3D COMPUTER GRAPHICS

In interactive computer graphics, semitransparent objects are usually simulated using a technique known as "*alpha blending*," first introduced by Porter and Duff[60] in 1984 and nowadays supported natively by the graphics hardware. In this technique, each *fragment* (Pixel prototype resulting from the rasterization of a geometrical primitive [e.g., a triangle]) has an associated *alpha* value representing its opacity. To "blend" a new fragment in the image under construction, a new color is computed as a function of the color of the fragment and of the one already present in the considered location of the frame buffer (Portion of the memory of the graphics card storing the image that will be displayed on screen) (resulting from the processing of the previously submitted fragments). Hence, the new color will replace the old one in the

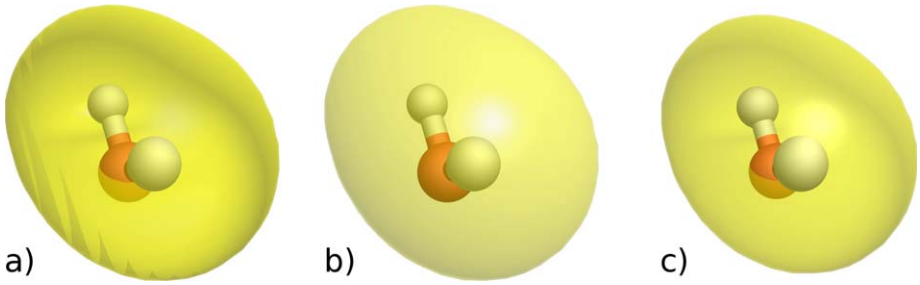

FIGURE 12 Comparison between different rendering techniques for simulating semitransparent surfaces. An orbital of a water molecule is shown. (a) Alpha blending of front and back faces produce graphical artifacts. (b) Alpha blending of front faces only removes (most of) the graphical artifacts but produce inaccurate images (e.g., the oxygen atom does not appear crossed by the orbital). (c) Image obtained using the "*Weighted Blended Order-Independent Transparency*" technique. Even if this is an approximate method, it produces plausible results. The orbitals have been computed at HF/STO-3G level of theory with Gaussian 09[56] suite of programs

frame buffer. The function usually chosen for this task is the following (named "*OVER*" operator in reference 60):

$$\mathrm{RGB}_{\mathrm{new}} = (\mathrm{RGB}_{\mathrm{frag}} \cdot A_{\mathrm{frag}}) + ((1 - A_{\mathrm{frag}}) \cdot \mathrm{RGB}_{\mathrm{old}})$$

Where $\mathrm{RGB}_{\mathrm{new}}$ is the color resulting from the blending operation, $\mathrm{RGB}_{\mathrm{old}}$ is the color previously stored in the frame buffer at the considered location, $\mathrm{RGB}_{\mathrm{frag}}$ is the color of the new fragment and $A_{\mathrm{frag}} \in [0, 1]$ represents the opacity of the fragment. The problem with the *OVER* operator is that the final color for a given pixel depends on the order in which the fragments laying on that pixel are blended. In other words, the rendered image can vary depending on the position of the virtual camera in the 3D scene, on the coordinates of the geometrical primitives in the 3D scene, and on the order in which these primitives are submitted to the rendering system. From the user's point of view this may result in incorrect colors and/or graphical artifacts, as shown in Figure 12a.

One solution to this problem consists in drawing the opaque polygons first (in any order), and then drawing the semitransparent polygons from the farthest to the nearest to the virtual camera (with the writing on the Z-buffer disabled). However, depending on the number of semitransparent polygons, ordering them every frame may not be a viable solution for a real-time interactive application. Furthermore, the problem persists in the case of overlapping polygons. A popular cheaper alternative consists in ordering only the semitransparent "*objects*" of the 3D scene (instead of the triangles they are formed by) and to draw only their polygonal faces that are oriented in the direction of the camera ("*backface culling*"). However, this method produces roughly approximate results (see Figure 12b), works only in the case of non-overlapped semitransparent objects and may not remove entirely the graphical artifacts. In conclusion, the only way to obtain an *exact* alpha blending, using the "*OVER*" operator and in the general case, is to blend the fragments in the correct order. Examples of GPU-accelerated techniques which achieve this goal are the "depth peeling" methods.[61–63] Other GPU-accelerated approaches renounce to produce exact results to obtain better performance, for example, by defining a different compositing operator which is commutative.[62,64,65] Several other OIT methods are present in the literature and a comprehensive survey has been published.[66]

In Caffeine, we employed an approximated method, the "*Weighted Blended Order-Independent Transparency*" by McGuire and Bavoil[64,67] (see Figure 12c), because it provides a good balance between quality of the results, performance, and implementation complexity.