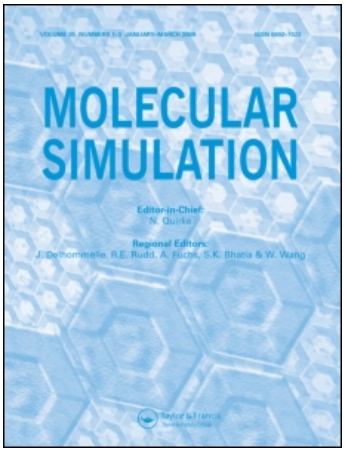

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

3D Visualization of Molecular Simulations in High-performance Parallel Computing Environments

Karsten Meier[a]; Christopher Holzknecht[a]; Stephan Kabelac[a]; Stephan Olbrich[b]; Karsten Chmielewski[b]
[a] Institute for Thermodynamics, University of the Federal Armed Forces Hamburg, Hamburg, Germany [b] Regional Scientific Computing Centre for Lower Saxony, University of Hannover, Hannover, Germany

To cite this Article Meier, Karsten , Holzknecht, Christopher , Kabelac, Stephan , Olbrich, Stephan and Chmielewski, Karsten(2004) '3D Visualization of Molecular Simulations in High-performance Parallel Computing Environments', Molecular Simulation, 30: 7, 469 — 477
To link to this Article: DOI: 10.1080/08927020410001680778
URL: http://dx.doi.org/10.1080/08927020410001680778

Molecular Simulation, Vol. 30 (7), 15 June 2004, pp. 469–477

Taylor & Francis
Taylor & Francis Group

3D Visualization of Molecular Simulations in High-performance Parallel Computing Environments*

KARSTEN MEIER[a,†], CHRISTOPHER HOLZKNECHT[a], STEPHAN KABELAC[a], STEPHAN OLBRICH[b] and KARSTEN CHMIELEWSKI[b]

[a]*Institute for Thermodynamics, University of the Federal Armed Forces Hamburg, Holstenhofweg 85, D-22043 Hamburg, Germany;*
[b]*Regional Scientific Computing Centre for Lower Saxony, University of Hannover, Schloßwender Str. 5, D-30159 Hannover, Germany*

(Received July 2003; In final form February 2004)

This paper presents a novel tool for interactive 3D visualization and computational steering of molecular simulations and other computer simulation techniques such as computational fluid dynamics in parallel computing environments. The visualization system consists of three major components—data source, streaming server and viewer—which are distributed in intra/internet networks. A parallelized data extraction and visualization library, which generates 3D scenes, is integrated in the simulation software. The 3D scenes can be stored locally or sent to the streaming server parallel to the simulation, where they are stored on a fast RAID hard disk system. The streaming server transfers the 3D scenes to viewer clients on demand for display. A multi-platform 3D viewer software is provided as a plug-in embeddable into different WWW browsers. Many models of broad interest in molecular simulations can be visualized, from simple spherical particles to moderately complex molecules, and several volume visualization methods are implemented efficiently. Examples from thermophysical property research applications demonstrate the utility of the visualization system. One example shows that the transport coefficients of the Lennard–Jones fluid at subcritical temperatures in the gas region are influenced by the formation of small clusters of particles.

Keywords: 3D visualization; Cluster formation; Lennard–Jones potential; Molecular simulation; Parallel computing

INTRODUCTION

3D visualization of molecular simulations can help to interpret simulation results and can contribute significantly to the understanding of mechanisms on the molecular scale, which are responsible for the macroscopic behavior of fluids. For example, when 3D scene sequences from simulations are visualized as smooth animations, the motion of individual particles or groups of particles can be followed easily, giving the simulator direct visual insights into the dynamics on the molecular scale. Molecular simulations employing a few hundred molecules of moderate complexity already require large computational resources. Therefore, they are usually carried out on high-performance parallel computers. The addition and integration of a visualization into the simulation software turns out to be a challenging task, especially if it is intended to build an efficient, balanced and scalable process chain and to add only a small additional computational cost on the simulation.

The aim of this paper is to provide a novel network-distributed visualization system for interactive 3D visualization of molecular simulations in high-performance parallel computing environments. Major differences of the system presented in this paper to other visualization systems are as follows: firstly, the software and the format of the data are optimized for speed and cover a wide range of 3D visualization primitives. Second, a streaming system enables us to explore 3D scenes while the simulation is in progress without access to local hard disks. Moreover, the system can be used in environments with a combination of shared and distributed memory architectures. Interactive navigation of

*Paper presented at FOMMS 2003—Foundations of Molecular Modeling and Simulation, July 6–11, 2003, Keystone, Colorado, USA.
†Corresponding author. Tel.: +49-0-40-6541-2161. Fax: +49-0-40-6541-2005. E-mail: karsten.meier@unibw-hamburg.de

ISSN 0892-7022 print/ISSN 1029-0435 online
DOI: 10.1080/08927020410001680778

3D scenes, computational steering, e.g. changing simulation parameters while the simulation is in progress, and collaborative exploration of molecular simulations by spatially separated spectators are supported.

CONCEPT

In general, the visualization process generates a sequence of images from simulation data in four stages. These four stages constitute the abstract visualization pipeline, which is shown in Fig. 1. First, a filtering process selects and modifies data generated by the simulation, e.g. the data source. Second, descriptions of graphics data are generated from the filtered data in a mapping process. Graphics data describe geometries and visual properties of the objects to be displayed. Subsequent rendering transforms the graphics data to an image. In the last display stage, the image is presented to the user. The system described in this paper is a software that implements the visualization pipeline [1,2]. For every stage in the pipeline, there is one component in the visualization system.

Four basic cases for a subdivision of the visualization pipeline can be distinguished as depicted in Fig. 1. Generally, at every step the data volume as well as the level of interaction control is reduced. The traditional approach of visualization in scientific computing involves storage and transfer of raw results and interactive exploration on a dedicated postprocessing system (A). In order to take advantage of hardware-supported 3D rendering, approach (C) was chosen, which supports client-side, interactive navigation of 3D scenes in a virtual reality environment. Compared with approaches (A) and (B), significant data compression is achieved, especially for 3D volume data visualizations.

The visualization system consists of three major components—data source, streaming server and viewer—which are distributed in high-performance intranet/internet networks among heterogeneous computers. The simulation software integrates a parallelized data extraction and visualization library, which maps time-dependent results into a sequence of composed 3D objects as abstract representations of the data. Parallel to the simulation, the generated 3D scenes can be stored locally or sent to the streaming server, where they are stored on a high-performance RAID hard disk system. The streaming server transfers the 3D scenes to viewer clients on demand for display. A multi-platform 3D viewer software is provided as a plug-in embeddable into different WWW browsers. The viewer component utilizes the OpenGL® application programming interface and can therefore exploit any hardware acceleration available on the client site. Basically, this realization of the visualization pipeline consists of two steps: efficient postprocessing of simulation data delivers 3D scenes, and buffering and viewing of the sequence of 3D scenes.

CREATION OF 3D SCENES

To enable interactive 3D exploration techniques at the client side, graphics primitives, such as polygons, lines or points in 3D space, are generated. 3D scenes are created to represent the characteristics of the simulation results. Typical examples for 3D visualization of three-dimensional data are isosurfaces, colored slicers, streamlines or, in the case of molecular simulations, sets of particles represented by spheres or other simple geometric bodies, which are composed in 3D scenes. Contrary to sequences of 2D images, sequences of 3D scenes allow the exploration of scenes from different viewpoints during the simulation, provide a flexible choice of rendering options and offer the advantages of real-time navigation techniques and interactive exploration of spatial data. For parameter variations, such as

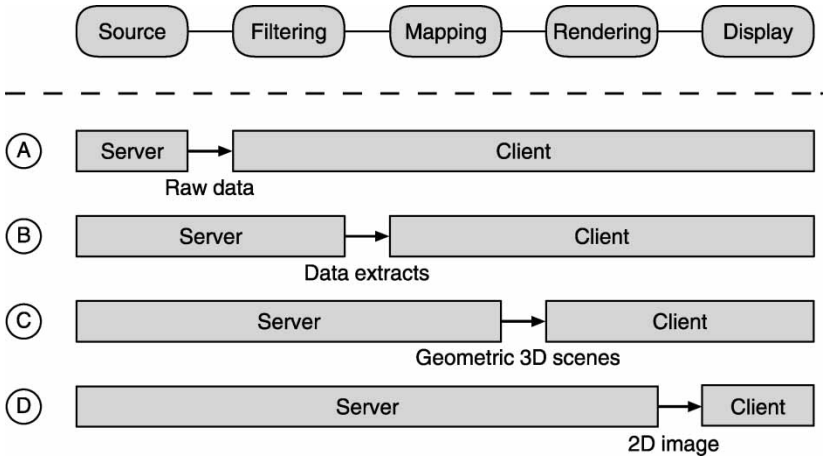

FIGURE 1 Different approaches of partitioning the visualization pipeline.

time steps in the simulation, a series of 3D scenes is generated resulting in a smooth 3D animation of the simulation results. The system renders the 3D animation frame by frame in parallel to the simulation. One frame corresponds to one simulated time step and contains a complete 3D scene.

To get high utilization of the computing and networking resources, a special binary representation for 3D scenes is defined, the DVR format. DVR is a data format to specify topology information, coordinates, normal vectors and other visual properties of 3D objects as well as defaults for virtual cameras and lighting. It is used for the 3D files and streams in the network-distributed system. By designing this special binary format, the client-side bottleneck from decoding, parsing, traversing, and optimization operations required for standard formats such as VRML is avoided. Furthermore, network-traffic and storage-requirements are reduced.

The mapping process of the raw simulation data, e.g. particle coordinates, into geometric 3D scenes, in which particles are for example represented by 3D spheres, is realized as an MPI-based Fortran-callable, portable DVRP library "libDVRP" in C, which is linked to the MPI-based parallel simulation software. An interface module for Fortran is provided for the benefit of modern Fortran 90/95 language constructs. In this way, it is possible to take full advantage of the parallelization strategy of the simulation software and apply the filtering and mapping algorithms to the partial results on every individual processor. The library provides parallelized visualization functions and, furthermore, it contains communication routines, which encapsulate recording of 3D scenes on a remote streaming server. A back channel is also integrated for computational steering of molecular simulations. Furthermore, a meta file format "DVRS" is used to describe a sequence of DVR files. Detailed descriptions of the library functions are available on the project WWW site [3].

STREAMING SYSTEM

With the streaming system, sequences of 3D scenes are efficiently stored and played out in a high-performance networking and visualization environment. The complete architecture of the system consists of the three components source, streaming server and viewer, which can be distributed over Internet protocol networks and operated either synchronously in pipelined mode or asynchronously. In synchronous pipelined mode, the streaming system enables us to explore simulation results, while the simulation is in progress. The source creates 3D scenes and sends them to the streaming server as described in the previous section. The streaming server provides buffer space resources, recording and play-out features, which are based on Real Time Streaming Protocol (RTSP) [4] for control purposes and the optimized transport protocol DVRP for 3D scenes both on top of TCP/IP. The streaming server supports RECORD and PLAY mode. In RECORD mode, the server receives and stores 3D scene sequences, which are handled as a media stream similar to recording mode on a video streaming server. On the other hand, in PLAY mode the server delivers a sequence of existing 3D scenes to the viewer either synchronously similar to live video streaming or asynchronously similar to video on demand in accordance to navigation commands from clients. The streaming server decouples the timing of the simulation time steps and the timing which is necessary for the presentation of 3D scene sequences as smooth animations, e.g. it enforces a fixed frame rate.

The viewer software "DocShow-VR" provides communication, presentation and interaction capabilities. It is written in C and translates the DVR 3D scene descriptions into OpenGL® instructions. The viewer is embeddable into WWW browsers as a plug-in and currently supports several Windows® and Unix® platforms and different browsers. 3D scene sequences can be displayed in mono- or stereoscopic mode. Stereoscopic viewing is possible by use of a stereo splitter with a pair of projectors and polarizing filter glasses. Moreover, 3D navigation devices such as a space mouse are supported.

The operating sequence of the streaming system is illustrated in Fig. 2. It is characterized by the following steps:

Source → streaming server

1. The simulation software includes a client (libDVRP), which connects to the streaming server and transmits control commands over RTSP.
2. A sequence of 3D scenes (DVR files) is transmitted over TCP/IP (DVRP).

Streaming server → viewer (*synchronously or asynchronously to step 1. and 2.*)

3. A DVRS meta file is fetched from a WWW server by the browser via HTTP.
4. Based on the location and attribute specifications in the DVRS meta data, connections to one or more streaming servers are established by the viewer plug-in.
5. The streaming server reads 3D scenes from files in DVR format and delivers them to the viewer client.

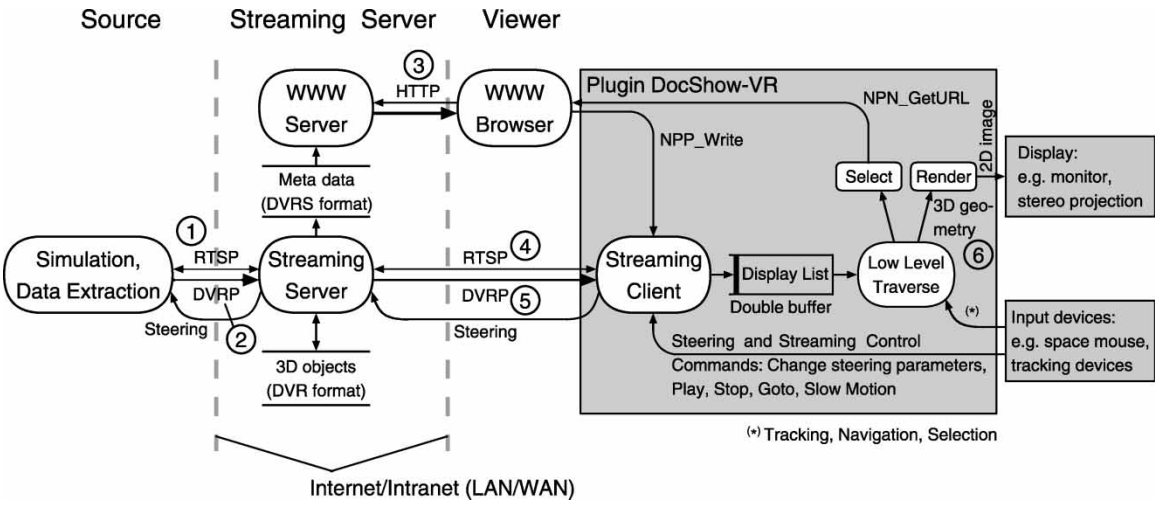

FIGURE 2 Communication between the network-distributed components generator, streaming server and viewer of the visualization system.

6. Further actions of the plug-in: reading 3D data and 3D rendering, 3D navigation and control of the streaming server according to user input via streaming and steering control.

Viewer → streaming server → source (*computational steering*)

By using this back channel, the process chain enables interactive computational steering of a simulation. It is based on RTSP commands and a bidirectional extension of the data connection (DVRP). The navigation and steering parameters can be controlled by graphical user interfaces—streaming and steering control—of the viewer.

Figure 3 depicts the streaming and steering control dialog windows.

Streaming

The maximum viewing frame rate is limited by the rendering time for the individual 3D scenes and the network bandwidth. The goal for the frame rate is user-selectable. However, if that goal cannot be met, the user can, via the "force speed" option, either decide to view the complete sequence at the expense of frame rate or allow an intermediate frame drop in order to keep the intra-stream synchronization. In the steering scenario, e.g. when simulation and playback are progressing simultaneously, progress-bars inform the user of the progress of both the playback and the recording sequence stream. Synchronicity between simulation and viewing can be ensured by the "track at end" option and is important in case the simulation

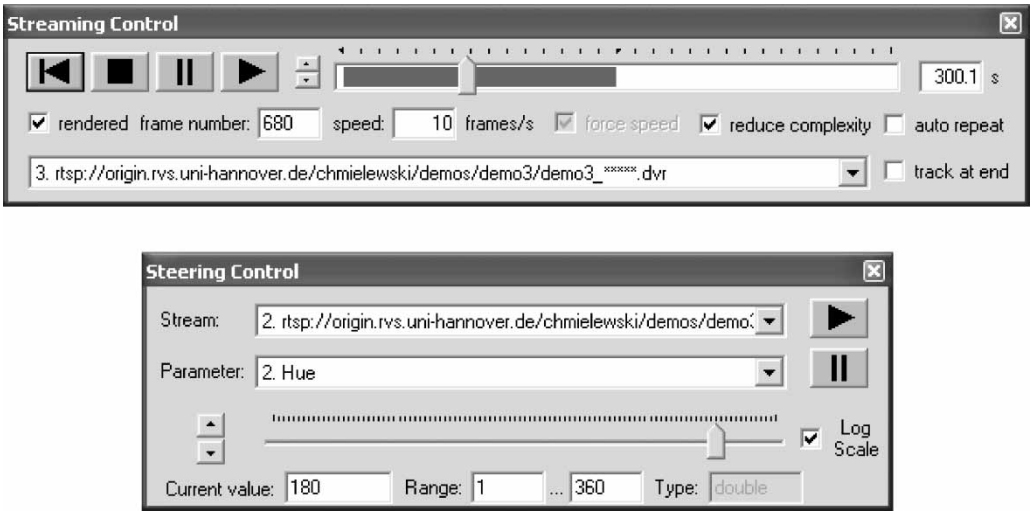

FIGURE 3 Streaming control and steering control dialog windows.

sequence progresses at a faster rate than play-back (inter-stream synchronization).

Steering

Parts of the steering control appear dynamically, as specified by the source client. Simulation parameters can be changed with sliders in the steering dialog window. In addition, the steering facility allows the user to suspend and continue the remote simulation process.

APPLICATIONS

With the visualization system, a range of models of broad interest in molecular simulations can be visualized. Furthermore, several other visualization methods, e.g. isosurfaces or slicers, are available. The library libDVRP contains visualization functions for displaying spherical particles in different colors and sizes. Moreover, spherical particles with embedded dipoles indicated by three-dimensional arrows can be visualized. Molecules consisting of several atoms can either be visualized by displaying all atoms separately, or object-orientated, by defining molecules as groups of spherical particles of different size and color in a rigid spatial arrangement and displaying these 3D objects according to their position and orientation in the simulation box. The visualization system was used in several applications of thermophysical property research, from which some examples are described in the remainder of this paper. The simulation software used in these case studies is based on FORTRAN listings provided as microfish attachments to the book by Allen and Tildesley [5] and was efficiently parallelized for distributed memory environments using MPI [6]. In the following discussion, reduced quantities denoted by an "$*$" are used if not denoted otherwise, e.g. $T^* = Tk/\varepsilon$, $\rho^* = \rho\sigma^3$ and $\mu^{*2} = \mu^2/4\pi\varepsilon_0\sigma^3\varepsilon$, where T denotes the temperature, k is the Boltzmann constant, ρ stands for the number density, μ is the dipole moment, ε and σ represent Lennard–Jones potential parameters and ε_0 is the permittivity of vacuum.

Cluster Formation in the Gas Region

In a recent simulation study [6,7], the transport coefficients viscosity, bulk viscosity and the self-diffusion coefficient of the Lennard–Jones fluid were determined in a large part of the fluid region of the phase diagram by equilibrium molecular dynamics simulations. Beside the transport coefficients, time-correlation functions of the thermodynamic fluxes were calculated at every simulated state point. In terms of the Green–Kubo formulas, the transport coefficients are related to the integrals of the time-correlation functions of the thermodynamic fluxes, which themselves are determined by the coordinates and velocities of the particles in

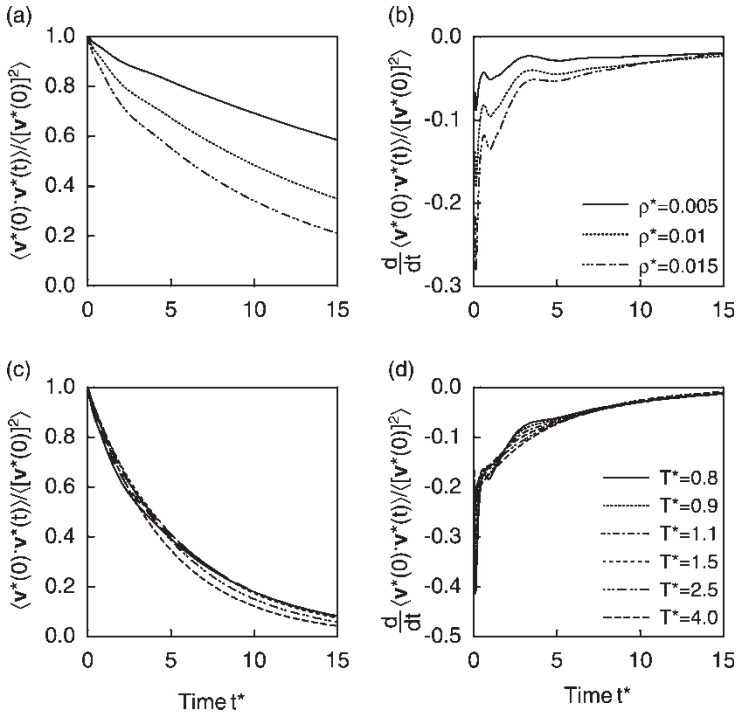

FIGURE 4 Short time behavior of the normalized velocity autocorrelation function and its time derivative in the gas region of the Lennard–Jones fluid. (a) and (b): Density dependence on the isotherm $T^* = 0.7$. (c) and (d): Temperature dependence along the isochor $\rho^* = 0.025$.

the system [5]. Thus, the decay behavior of the time-correlation functions provides insights into the transport mechanisms on the molecular scale. This connection can be most directly explored for the self-diffusion coefficient because the corresponding thermodynamic flux is the velocity vector of a single particle. Figure 4 shows simulation results for the velocity autocorrelation function at sub- and supercritical temperatures in the gas region. At low temperatures, the decay of the correlation functions is superimposed by small oscillations, which decay after a few cycles. The oscillations are clearly evident in the derivatives of the correlation functions. The strongest effect is observed for the lowest temperature $T^* = 0.7$ close to the triple point temperature of the Lennard–Jones fluid. With increasing temperature, the oscillations become smaller and vanish for the highest temperatures displayed in Fig. 4. The oscillations were also observed for shear stress and pressure fluctuation correlation functions [6]. It is assumed that the oscillations are due to the formation of clusters at low temperatures.

Similar observations were reported by Michels and Trappeniers [8,9] for velocity autocorrelation functions of the Lennard–Jones and square well fluid and for shear stress correlation functions of the Lennard–Jones fluid at low temperatures calculated by molecular dynamics simulations. Marchetti and Dufty [10] calculated the short time behavior of the velocity autocorrelation function at low densities by means of kinetic theory for the square well fluid. In their theory, the correlation function was separated into contributions due to scattering and bound states. The bound state contribution showed damped oscillations, whereas the scattering contribution decayed monotonically. The total velocity autocorrelation function was in excellent agreement with the simulation results of Michels and Trappeniers [8]. The occurrence of the oscillations can be explained by using that the velocity autocorrelation function is the scalar product of the velocity vector at the time origin and the velocity vector at a later time *t*and, thus, closely related to the angle between the two vectors. For example, in the case of dimers, e.g. clusters of two particles, internal vibrations and rotations of the dimer relative to its centre of mass introduce periodic components into the motions of the particles that superimpose the translational motion of the centre of mass of the dimer. In an undisturbed vibrating dimer, the relative velocity of the two particles is reversed within half an oscillation period resulting in a negative contribution to the velocity autocorrelation function. In the following half period, the relative velocities return to their initial value, yielding a positive contribution. Due to the permanent creation and destruction of bound states by collisions with other particles and the existence of a whole spectrum of rotational and vibrational frequencies, the oscillations in the velocity autocorrelation functions are damped out rapidly.

This effect provides an excellent example to be investigated by 3D visualization. When particles belong to a cluster, free particles are displayed by different colors: the clusters can then be easily identified and their motion can be followed over many time steps. The definition of a cluster is ambiguous—see for example the discussion in the context of percolation phenomena in the supercritical region of the Lennard–Jones fluid by Sator [11]. In this work, bound pairs of particles are identified by the criterion used by Rainwater in the theory of the second transport virial coefficients [12]. If a particle belongs to several bound states and the bound particles themselves form bound states with further particles, all bound particles form one large cluster.

There are two types of bound particles: metastable and stable bound particles. Two isolated particles form a stable cluster if their relative kinetic energy in the center of mass frame of the two particle system is smaller than the well depth of the Lennard–Jones potential, e.g. if the total energy of the two particle system in the centre of mass frame is negative. When the total energy is positive, two particles can also form bound states if they are bound by the centrifugal barrier. The relatively complicated derivation of this criterion is described by Rainwater [12]. Due to the possibility of tunnelling, these clusters are metastable if the system is described by quantum mechanics, but they are stable if the system is described by classical mechanics. Although all systems are treated by classical mechanics in this work, clusters of this type are termed metastable to distinguish them from the truly stable clusters. Strictly, this cluster definition applies only to two isolated particles. However, it was observed that clusters identified by this criterion were stable over many time steps.

For the identification of clusters the algorithm described by Swope *et al.* [13] was implemented into the simulation software, and two steering channels were added to enable changes of the temperature and density during the simulation. Several simulations at low temperature states in the gas region of the Lennard–Jones fluid were carried out to produce 3D animations with the visualization system. Particles belonging to stable and metastable clusters were displayed by different colors and their motion was inspected. The expected internal vibrations and rotations of the cluster could indeed be observed. Figure 5 shows a 3D scene from a simulation of the Lennard–Jones fluid at the state point

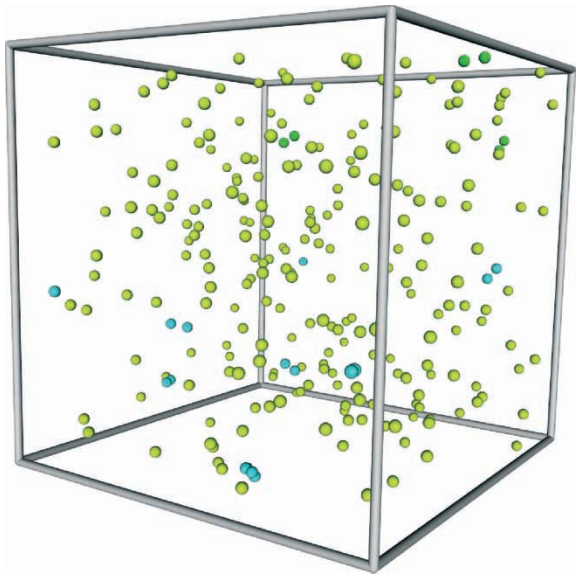

FIGURE 5 3D scene from a molecular dynamics simulation of the Lennard–Jones fluid with 256 particles at the state point ($T^* = 0.9$, $\rho^* = 0.005$). Stable bound particles are displayed in blue, metastable bound particles in green and free particles in yellow colour.

($T^* = 0.9$, $\rho^* = 0.005$). Typically, the visualization overhead of a simulation with up to 1372 Lennard–Jones particles in the gas region including the cluster analysis amounted to less than 1% of the total CPU-time when the number of processors is chosen optimally for the best parallel performance of the simulation software.

In a further investigation, it was ascertained whether clusters also exist in the gas region of the dipolar Stockmayer fluid, whose particles interact with a Lennard–Jones potential plus an embedded point dipol. Several molecular dynamics simulations were performed at the density $\rho^* = 0.01$ and different subcritical temperatures. Figure 6 shows a 3D scene from a simulation at the state point ($T^* = 1.085$, $\rho^* = 0.01$). Several clusters consisting of two or more particles were observed. Furthermore, the animation revealed that the internal motion of the clusters is much more complex than for the Lennard–Jones clusters as could be expected from the strongly orientation dependent interactions of the Stockmayer potential.

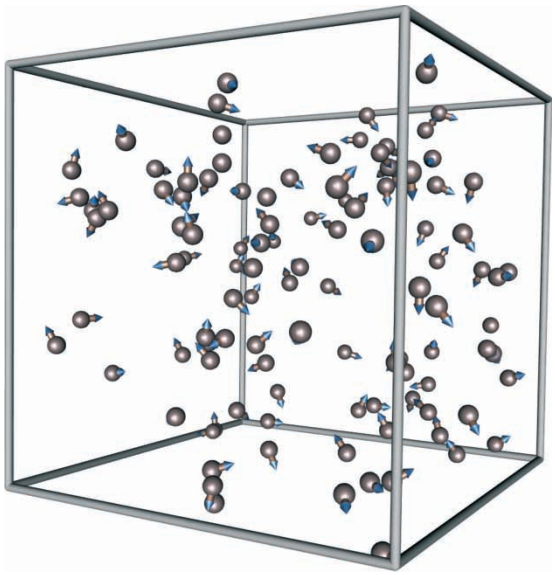

FIGURE 6 3D scene from a molecular dynamics simulation with 108 Stockmayer particles with dipole strength $\mu^{*2} = 3.0$ at the state point ($T^* = 1.085$, $\rho^* = 0.01$).

Further Examples

Planar vapor–liquid interfaces are often simulated to determine the surface tension [14]. At the start of the simulation, a cubic box containing particles at a liquid density is placed between two empty boxes and the usual periodic boundary conditions are applied to this new box. After an equilibration phase, two stable vapor–liquid interfaces are formed with the liquid phase in the centre third and two gas phases in the left and right third of the box. Figure 7 depicts a 3D scene from a simulation of the vapor–liquid interface of the Lennard–Jones fluid at the temperature $T^* = 1.1$. This and further simulations at other temperatures were carried out to calculate the surface tension and profiles of the density, temperature and tangential and normal pressure components in the interfacial region. 3D visualizations of these simulations were found to be useful to follow the equilibration and to gain insights into the processes in the interfacial region.

For molecular dynamics simulations of molecules consisting of several atoms, the choice of the visualization method depends on different aspects such as the method employed to integrate the equations of motion or the coordinate system used to represent the molecular geometry. In any case, molecules can be visualized by displaying all atoms individually as spheres of different size and color. This approach is advantageous if constraint dynamics is used to integrate the equation of motion because the positions of the atoms within a molecule often coincide with the interaction sites or can easily be related to them. The object-orientated approach is preferable if the quaternion representation of the equations of motion is used since the orientations of the molecules are known at every time step. Figure 8 shows a 3D scene from a simulation of liquid water at ambient conditions. The interactions between water molecules were described by the SPC/E model [15] and the equations of motion were solved using constrained dynamics by the RATTLE algorithm [5].

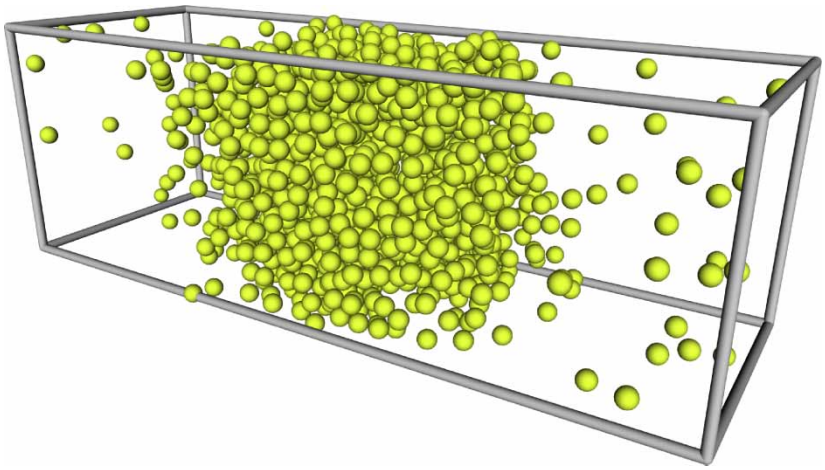

FIGURE 7 3D scene from a molecular dynamics simulation of the planar vapour–liquid interface of the Lennard–Jones fluid with 1372 particles at the temperature $T^* = 1.1$. (Colour version available online.)

3D animations of the case studies described in this paper and further examples can be viewed on the project WWW site [3]. Furthermore, several examples for 3D volume visualizations from meteorological and fluid mechanical applications are available on the project WWW site. Although only visualizations of molecular dynamics simulations have been discussed, the visualization system can also be applied to produce animations of Monte Carlo simulations. Besides the examples presented here, the system is capable of visualization of more complicated molecular models and can be applied to visualize many different physical situations and phenomena commonly investigated by molecular simulations.

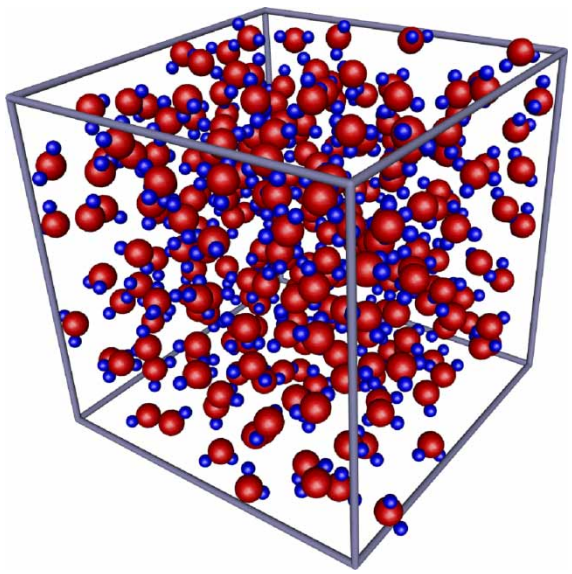

FIGURE 8 3D scene from a molecular dynamics simulation of liquid water with 256 molecules at the state point ($T = 300\,\mathrm{K}$, $\rho = 998\,\mathrm{kg/m^3}$). (Colour version available online.)

CONCLUSIONS

A distributed system for 3D visualization of molecular simulations in parallel computing environments was successfully implemented. It provides an efficient method to explore simulation results in a high-performance networking and 3D graphics environment with only slight additional computational load on the simulation. It enables smooth animations of molecular simulations with stereoscopic viewing options and interactive 3D navigation techniques. Furthermore, simulation steering capabilities were added, based on a back channel in the data connection. The utility of the visualization system was demonstrated by several applications from thermophysical property research. Besides, the system can be used in other computer simulation methods, e.g. computational fluid dynamics, to visualize 3D volume data.

Acknowledgements

This project was part of a gigabit (LAN/WAN) testbed project in the DFN (German Academic Network) and financially supported by the German Federal Ministry of Education and Research (BMBF). Computational resources were provided by the Regionales Rechenzentrum für Niedersachsen (RRZN) at the Universität Hannover and the Konrad-Zuse-Zentrum für Informationstechnologie (ZIB) in Berlin.

References

[1] Olbrich, S., Pralle, H. and Raasch, S. (2001) "Using streaming and parallelization techniques for 3D visualization in a high-performance computing and networking environment", In: Hertzberger, B., Hoekstra, A. and Williams, R., eds, *Lecture Notes in Computer Science* (Springer Verlag, Berlin), pp 231–240.

[2] Jensen, N., Olbrich, S., Pralle, H. and Raasch, S. (2002) "An efficient system for collaboration in tele-immersive

environments", In: Bartz, D., Pueyo, X. and Reinhard, E., eds, *Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization* (Eurographics Association, Blaubeuren), pp 123–131.
[3] http://www.rvs.uni-hannover.de/projekte/tele-immersion.
[4] Schulzrinne, H., Rao, A. and Lanphier, R. (1998) "RFC 2326 — Real Time Streaming Protocol (RTSP)", Internet RFC/STD/FYI/BCP Archives, http://www.faqs.org/rfcs/rfc2326.html.
[5] Allen, M.P. and Tildesley, D.J. (1987) *Computer Simulation of Liquids* (Clarenden Press, Oxford).
[6] Meier, K. (2002) *Computer Simulation and Interpretation of the Transport Coefficients of the Lennard–Jones Model Fluid* (Shaker Verlag, Aachen).
[7] Meier, K., Laesecke, A. and Kabelac, S. (2001) "A molecular dynamics simulation study of the self-diffusion coefficient and viscosity of the Lennard–Jones fluid", *Int. J. Thermophys.* **22**, 161–173.
[8] Michels, J.P.J. and Trappeniers, N.J. (1975) "Molecular-dynamical calculations of the self-diffusion coefficient below the critical density", *Chem. Phys. Lett.* **33**, 195–200.
[9] Michels, J.P.J. and Trappeniers, N.J. (1981) "Molecular dynamical calculations on the transport properties of a square well fluid. IV. The influence of the well-width on the viscosity and the thermal conductivity", *Physica A* **107**, 299–306.
[10] Marchetti, M.C. and Dufty, J.W. (1981) "Bound-state and finite-collision-time effects in the binary-collision approximation", *Phys. Rev. A* **24**, 2116–2134.
[11] Sator, N. (2003) "Clusters in simple fluids", *Phys. Rep.* **376**, 1–39.
[12] Rainwater, J.C. (1984) "On the phase space subdivision of the second virial coefficient and its consequences for kinetic theory", *J. Chem. Phys.* **81**, 495–510.
[13] Swope, W.C., Andersen, H.C., Berens, P.H. and Wilson, K.R. (1982) "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters", *J. Chem. Phys.* **76**, 637–649.
[14] Nicolas, J.P. and Smit, B. (2002) "Molecular dynamics simulations of the surface tension of n-hexane, n-decane and n-hexadecane", *Mol. Phys.* **100**, 2471–2475.
[15] Berendsen, H.J.C., Grigera, J.R. and Straatsma, T.P. (1987) "The missing term in effective pair potentials", *J. Phys. Chem.* **91**, 6269–6271.