Vision, Modeling, and Visualization (2011)
Peter Eisert, Konrad Polthier, and Joachim Hornegger (Eds.)

Interactive Exploration of Polymer-Solvent Interactions

B. Thomaß[1], J. Walter[2], M. Krone[1], H. Hasse[2], and T. Ertl[1]

[1]Visualization Research Center (VISUS), University of Stuttgart, Germany
[2]Laboratory of Engineering Thermodynamics, University of Kaiserslautern, Germany

Abstract

The interaction of three-dimensional linked hydrophilic polymers with surrounding solvents in time-dependent data sets is of great interest for domain experts and current research in molecular dynamics. These polymers are called hydrogels, and their most characteristic property is their swelling in aqueous solutions by absorbing the solvent. Their conformation transition can be studied by investigations of the interaction of the single polymer strand and the solvent directly around the polymer at an atomistic level. We present new visualization techniques to interactively study time-dependent data sets from molecular dynamics simulations—with special regard to polymer-solvent interactions like local concentrations and hydrogen bonds—as well as filtering methods to facilitate analysis. Such methods that visualize polymer-solvent interactions on a hydration shell around a polymer are not available in current tools and can greatly facilitate the visual analysis, which helps domain experts to extract additional information about hydrogel characteristics and gain new insights from the simulation results. While our visual analysis methods presented in this paper clearly facilitate the analysis of hydrogels and lead to new insight, the presented concepts are applicable to other domains like proteins or polymers in general that interact with solvents.

Categories and Subject Descriptors (according to ACM CCS): J.2 [Computer Applications]: Physical Sciences and Engineering—Chemistry I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary Representations I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing I.4.10 [Image Processing and Computer Vision]: Image Representation—Volumetric

1. Introduction

The focus of this work lies in the visualization of special polymers which are part of a hydrogel and their interaction with the surrounding solvent. The molecular dynamics simulation of this scenario leads to some fundamental questions in which our special visualization methods can help getting a better insight into the simulation results. In our simulation data many solvent atoms around the polymer make it hard to analyze the data by just rendering all atoms using standard tools. Therefore, we introduced spatial filtering of the solvent molecules to reduce visual clutter. Furthermore, hydrogen bonds between the polymer and the remaining solvent molecules are computationally extracted and visualized. In addition, the concentration of solvent molecules around the polymer is relevant for further analysis. We adressed this by mapping information about molecules in the hydration shell around the polymer onto a molecular surface. Since the domain experts are interested in dynamic features that change over time, we added the capability to accumulate values or show statistics over the trajectory.

1.1. Chemical background

Hydrogels are three-dimensional hydrophilic polymer networks. Their most characteristic property is their swelling by absorbing a solvent. Hydrogels can be used in many applications like superabsorbers [ER05] (e.g. diaper fillings) and contact lenses [PSI*09]. The swelling is influenced by various factors [HXM06]. On the one hand, the swelling depends on the structure of the hydrogel itself. On the other hand, environmental conditions such as temperature and solvent influence the swelling as well. Upon varying these factors, there is a parameter scope in which the hydrogel is swollen and a scope in which it is collapsed. In between those two parameter scopes lies the region of conformation transition. To fully exploit the potential of hydrogels in their various applications, it is important to understand,

DOI: 10.2312/PE/VMV/VMV11/301-308

describe and predict their swelling behavior. These phenomena are still not fully understood and remain a current topic in application domain research. Molecular dynamics simulations are a valuable method for finding the desired insights on the behaviour of hydrogels [WDVH10, WEVH10]. These simulations with atomistic resolution yield an extensive amount of information. Visualization on the molecular scale—enhanced by additional analyses and visual cues—can contribute greatly to the understanding of these processes.

In this paper, the swelling of poly(N-isopropylacrylamide) (PNIPAAm) hydrogels in mixtures of water and methanol was studied by molecular dynamics simulations and analysed in detail by our novel visual analysis techniques.

Upon varying the methanol concentration these hydrogels show an unexpected behaviour with two volume transitions: they are swollen in pure water, collapse at comparatively low methanol concentrations and then swell again as the methanol concentration is further increased. This behaviour is discussed in detail in scientific literature but not yet fully understood [TKKW09, CSW06, MSS*93, SMT91, WHV90, OS07]. Here, topics like the number of hydrogen bonds or the interactions between the solvent and the hydrogel or between the two solvents are discussed, comparing the stretched and collapsed state of the hydrogel. The results obtained using our methods presented in this work yield a satisfactory explanation for that behaviour. These methods employ advanced visual analysis of the structure of the solvent around the polymer strand in terms of local compositions, orientation and hydrogen bonding.

This work will give more detailed insight into the structure and the behaviour of the solvent and the hydrogel. For these investigations, single PNIPAAm polymers with 30 monomers were used in the molecular dynamics simulations. Therefore, new visual analysis methods were designed and integrated in our visualization application, which is already used by our project partners.

2. Related Work

In available and commonly used visualization tools for molecular simulations like VMD [HDS96], UCSF Chimera [PGH*04] and PyMOL [DeL02], it is possible to view the hydrogen bonds or trajectory of the molecules of the molecular simulation. Furthermore, it is possible to compute spatial distribution functions for the visualisation of solvent around a macromolecule. In combination with the calculation of radial distribution functions, these tools lead to some insight into the structure of the solvent surrounding the polymer. In order to look at the solvent surrounding the polymer strand, it is however necessary to identify each solvent molecule and use its ID number in the visualization program. When using this approach, the solvent molecules leaving the shell around the solute are still shown and new ones have to be identified manually. Instead, an automatically filtering for the solvent around the polymer would be of great benefit. Furthermore, it is not possible with these tools to visualize the local solvent concentration at the surface of the solute. Hydrogen bonds can be visualized in available postprocessing tools for each timestep but they lack filtering and statistical information.

Nowadays, simple molecular models like ball-and-stick, which consist only of quadratic surfaces, are usually rendered using GPU-ray casting [SWBG06, LVRH07], which was originally introduced by Gumhold [Gum03] and Klein et al. [KE04] . Since then, various improvements have been presented, e. g. composed surfaces [RE05, GRE09] or higher order polynomial surfaces [LB06, dTLP07].

Molecular surfaces are commonly used to depict phenomena like interactions of molecules—called solute—with a specific solvent. The Solvent Excluded Surface (SES) [Ric77] is most prominently used for the visualization of the solute in this application, since it illustrates the boundary which separates the solvent from the solute. However, the analytical computation of the SES is computationally demanding [Con83]. Recently, approaches to compute the SES interactively and visualize it using GPU-ray casting were presented [KBE09, LBPH10]. This enables the maintenance of the SES for large, dynamic molecules.

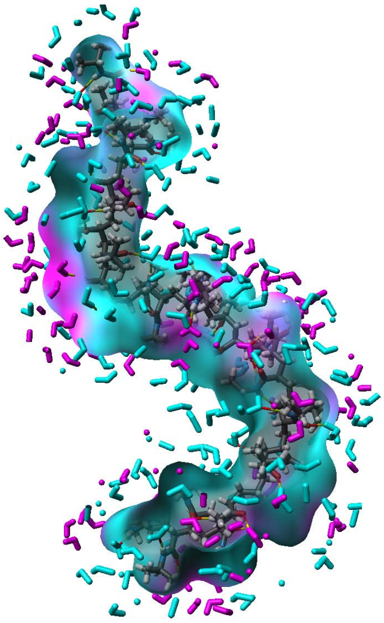


Figure 1: *Solvent concentration mapped onto a semi-transparent iso-surface of the polymer using color-coding for the competing solvent types (magenta for methanol, cyan for water). Only solvent molecules near the hydration shell of the polymer are visible.*

Protein-solvent interactions and visualization have been addressed by Bidmon et. al [BGB*08] with respect to solvent pathlines near protein cavities: however, this technique only allows analysis of solvent motion in limited, user-defined regions such as near protein binding sites.

The extraction and rendering of surfaces from volumetric data sets has a long tradition, most prominently in medical imaging. GPU-based ray casting [SSKE05, HLSR08] allows the rendering of smooth isosurfaces of high visual quality interactively on current graphics hardware. Volumetric surfaces were shown to be useful for particle-based data such as Smoothed Particle Hydrodynamics, where densities are computed by voxelizing the particle data [MCG03]. Recently, Giard et al. [GM10] and Dias et al. [DBG10] applied this to the field of molecular graphics. However, the Marching Cubes algorithm [LC87] which they applied to extract the surface causes visual artifacts due to the tessellation. Furthermore, for the interactive visualization of dynamic data sets, a recomputation of the volumetric representation per frame is crucial, which is not possible with their implementations in terms of speed. A fast, GPU-based volume generation technique was presented by Kolb and Cuntz [KLR04]. A similar approach was used by Falk et al. [FGE10] for cellular signal transduction visualization. Krone et al. [?] used this technique to visualize molecular surfaces and to find cavities within proteins. For the rendering, they all employ GPU ray casting.

3. Visualization methods

Based on the data from molecular dynamics simulation, we want to visually emphasize all relevant interaction of a polymer with its surrounding solvent over time. The first approach is to show the polymer atoms and the neighboring solvent atoms with a certain distance to the polymer. The atoms and their covalent bonds are shown using a ball-and-stick (or licorice) representation.

Futhermore, our application shows additional information color-coded onto a semi-transparent molecular surface around the one or more molecules defined as polymer. With this color-coding around the molecule we can visualize relevant information like concentration and type of surrounding solvent (Figure 1), highlight hydrogen bonds between solvent and polymer (Figure 2) or show statistics of hydrogen bonds (Figure 3). The user can switch between these available coloring modes in real-time.

The surface mentioned above is an approximation of the solvent excluded surfaces (SES) of the polymer. Calculating the actual SES for a given molecule is a non-trivial and expensive task. In our application we extract an iso-surface of a density volume generated by rendering atoms' positions with a smoothing kernel into a 3D-texture. The result is a good approximation which is sufficient for the analysis we want to facilitate and interactively adjustable.

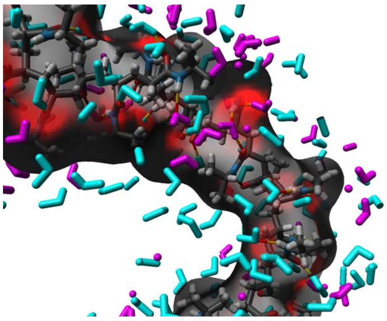


Figure 2: *Hydrogen bonds between polymer and solvent molecules rendered as thin cylinders. Their position is highlighted on the iso-surface (red).*

Hydrogen bonds between the polymer and the surrounding solvent are of special interest, because they play a key role in the chemical properties of a polymer. Our application identifies the hydrogen bonds and shows them as thin connections between molecules. In order to facilitate tracking of these hydrogen bonds it is possible to switch to a color-mapping mode where the hydrogen bonds are shown on the iso-surface of the polymer (Figure 2). This visualization method helps to track relevant hydrogen bonds over the trajectory, since the domain expert can follow highlighted regions on the semi-transparent polymer iso-surface, rather than searching for connections in a ball-and-stick representation only.

To identify hydrogen bonds in our data set we, have to find possible donor and acceptor atoms within a distance of approximately 3.5 Å of each other. In our case oxygen and nitrogen can be either acceptor or donor if they are covalently bonded to a hydrogen atom. In addition, the covalent bond between the hydrogen atom and the donor must be within a 30° angle of the connection from the donor to the acceptor (see Figure 4). In our application scenario we are only

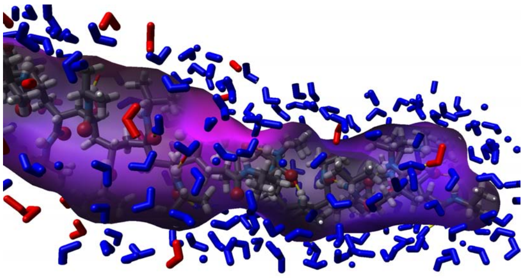


Figure 3: *Hydrogen bond statistics visualized by the sphere-radius of the atoms (more hydrogen bonds during the trajectory result in larger spheres) and color-coding (interpolation between the red and the blue solvent).*

interested in hydrogen bonds between the polymer and the surrounding solvents or intra-molecular hydrogen bonds of the polymer—not among the solvent molecules themselves.

In order to show statistical information about hydrogen bonds, our program collects statistics over the entire trajectory. The lifetime of each hydrogen bond is accumulated per polymer atom. Additionally, the type of the bonded solvent molecule is stored. The final representation increases the sphere radius of the atoms to visualize the likeliness of hydrogen bonds formed over the trajectory, based on the frequency and lifetime of the bonds. Color-coding onto the iso-surface is used to show what kind of solvent molecules form the hydrogen bonds to each polymer atom (see Figure 3). This visual information gives the user the chance to study how two competing solvent types interact with the polymer. One can observe if one solvent type becomes the dominant solvent in time or locality.

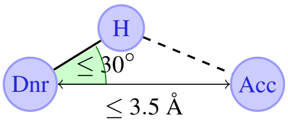


Figure 4: *Hydrogen bond criterion*

Another statistic which is computed in real time is the concentration of solvent molecule types in the hydration shell. For each frame, the number of molecules is extracted for all solvent types (in our case water and methanol). The user gets the exact number of molecules of each solvent type and can compare them to the global mixture ratio in the simulation. While these numbers lack the spatial information of the visualization where the location of the solvent molecules in the hydration shell is visible, the exact numeric mixture ratio provides additional information about the concentration of the different molecule types.

Since we are working with time-dependent data sets, we developed another concept of visually representing time-based correlations by accumulating values inside the volume which is used for generating the iso-surface around the polymer. When animating the trajectory, the 3D-texture for our volume is recomputed for every frame based on interpolated atom positions. If we add up the density values inside the volume for consequtive frames, we get the volume that the molecule occupies over time (Figure 5, top). This time-dependent accumulation can be done for colors inside the volume as well. In combination with selecting the coloring mode, users can choose from a wide varity of visual styles to assist the analysis (e.g. Figure 5, bottom). The user can toggle the accumulation and animation manually and can adjust the accumulation rate of each frame.

Animation of the trajectory is done with respect to the current frame rate and can be completely controlled by the user. For every frame all atom-positions are interpolated and the iso-surface around the polmyer is reconstructed in real time. Hydrogen bonds are recomputed per time-frame in the dataset and their lifetime clamped to $t = \pm\frac{1}{2}$ where t is the distance between the current animation time and the two nearest time-frames in the dataset. The CPU-code is parallelized to utilize multi-core systems.

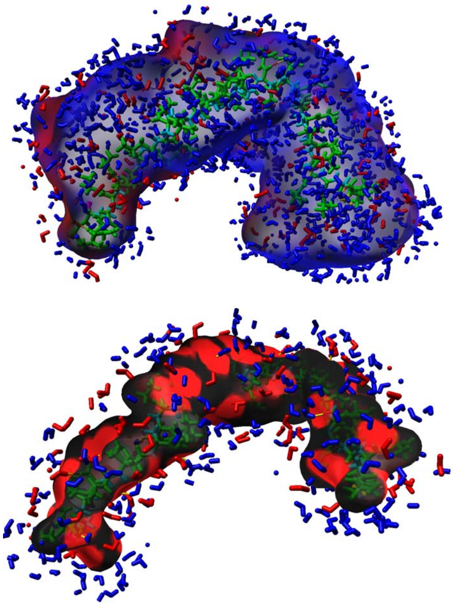


Figure 5: *Top: accumulation of the solvent surface around the polymer over a time period. Bottom: color-accumulation (hydrogen bonds coloring mode) over a time period.*

4. Implementation Details

The input data for our methods is a set of atoms and their corresponding positions in space over time as well as their element type. These atoms are grouped into a polymer and the different solvents. Covalent bonds are given implicitly by spatial neighbor relations and can be derived from the overlap of Van der Waals radii between atoms. If the radii of two atoms overlap by a given factor ($\sim \frac{2}{3}$), we can safely assume a covalent bond between these atoms is present. Since no chemical reaction takes place in the simulation, one can safely assume that all covalent bonds stay present during the entire trajectory—meaning that they can be determinated once in a pre-computation step, which speeds up the visualization.

4.1. Glyph-based Rendering

The rendering of the actual molecules uses the wide spread GPU-based glyph ray casting approach. Vertex-attributes per atom and bond, such as position, orientation, radius and color, are transferred to the graphics hardware as attributed point primitives. The molecules are rendered using spheres for atoms and cylinders for covalent bonds. The position of all atoms and bonds are sent as attributed `GL_POINTS` to the graphics card and rendered using the glyph-based GPU ray casting approach. The technical details can be found in the previous work from this area discussed in Section 2. This approach saves considerable CPU-GPU memory transfer bandwidth and improves visual quality compared to traditional triangle mesh techniques.

4.2. Hydrogen Bonds

The hydrogen bonds are calculated interactively for each snapshot of the trajectory. To speed up the process, we use a grid-based neighborhood-search of atoms in a given distance in space. Since atoms cannot be packed infinitely densely, a grid-based neighbor-search can be done in linear time, i.e. $\mathcal{O}(n)$. Because of the introduced criterion in Section 3, it is sufficient to perform a neighborhood search only on possible donor/acceptor atoms—oxygen and nitrogen in our case. Since we only want to get the bonds between the polymer and the solvent and inside the polymer itself, we perform a neighborhood search for every oxygen and nitrogen atom of the polymer within a distance of 3.5 Å. If we find other donor/acceptor atoms within range, we check whether there is a hydrogen atom covalently bonded to the donor that matches our hydrogen bond criterion (see Figure 4).

Since finding neighbors in the existing data set can be done in parallel, we use OpenMP to speedup the hydrogen bond search so it has a relatively small impact on the frame rate of the visualization (see Section 5).

Just like the atoms and their covalent bonds, we render the hydrogen bonds using GPU-based glyph ray casting. They are modeled as cylinder primitives with a smaller radius (a third of the covalent-bond radius) than the covalent bonds.

4.3. Molecular Surface Rendering

For the rendering of the color-coded polymer surface, we opted for a volume visualization approach. As mentioned in Section 2, the interactive computation and visualization of analytically defined smooth molecular surfaces like the SES are demanding and only possible for medium-sized molecules. Using interactive volume generation and rendering—for example [FGE10, **?**]—we can easily maintain the molecular surface for dynamic data in real time. Furthermore, the iso-surface allows us to easily project additional attributes to it as colors.

We create 3D-texture storing density values in the alpha channel as basis for the iso-surface rendering. The data we want to project onto the iso-surface is encoded to colors and stored in the RGB-channels of the corresponding voxels. The process to create the 3D-texture consists of two passes, where the first pass writes the density values and the second pass writes the colors. Generally, the volume generation follows the approach presented in [KC05], which can be outlined as follows:

The polymer atom positions will be splatted as point primitives into the corresponding slices of the 3D-volume texture using GLSL shaders and the frame-buffer-object mechanism. The radius of the atom and the size of the filter kernel determine the point size of the splat. The point size is set in the GLSL vertex program using the `gl_PointSize` command. The density values are accumulated into the 3D-texture using an additive blending operator. Additionally, the radial-symmetric smoothstep function is used to define a distance-based fall-off for the density.

In the second pass we accumulate color values from the surrounding solvent atoms into the 3D-texture using the same kernel function as for the density values. That is, data points are projected to a position and an influence radius and the selected color is written to the 3D-texture as color information for the surface. For example, when coloring by hydrogen bond, a data point is generated for each hydrogen bond center and the influence radius is set to half of the hydrogen bond length.

Please note that the second pass for coloring the volume can be accelerated using the density already accumulated in the volume 3D-texture. Atoms can be skipped in the vertex shader stage if the density is zero at their position. This saves a lot of fragment shader processing power and thus results in increased frame rates.

Now we can use the generated volume 3D-texture to visualize the iso-surface using GPU volume ray casting [HLSR08]. If we choose the iso-surface threshold and filter-radius properly we can obtain a molecular surface which approximates the SES of the polymer with a color-coding of the current surrounding solvent concentrations (Figure 1). In our experiments, a filter radius of 3.0 and an iso-value of 3.25 resulted in the desired appearance. Due to the ray casting, the iso-surface can be rendered semi-transparently with correct depth ordering. As depicted in the screenshots (i.e. Figure 1, 3), other representations such as the stick model can be combined correctly with the iso-surface.

The accumulation of values inside the volume as seen in Figure 5 is implemented using the OpenGL color mask mechanism. When the trajectory is animated, the volume must be refreshed every frame. We use `glColorMask` to clear the 3D-texture selectively depending on whether we want to accumulate density or color values or nothing or both. The rate of accumulation per frame can be directed by the user with a scalar value which is applied in the fragment processing stage of the volume generation.

4.4. Polymer-Solvent Rendering

Since we want to render the polymer molecules and some solvent molecules in the proximity of the polymer iso-surface, we have to render the solvent atoms with a distance-criterion to the polymer. Because we already have the 3D-volume-texture generated for the polymer iso-surface (see Section 4.3), we can use this volume instead of an expensive neighborhood-search for every solvent atom. If we look up the volume texture with the current vertex position in the vertex shader when rendering solvent atoms, we can skip vertices that are below a given threshold. If this threshold is lower than the iso-value of the polymer surface in Section 4.3, we render only solvent atoms in the proximity of the polymer iso-surface. The threshold-value for skipping solvent atoms is typically a number near zero (e.g. ≤ 0.01), so we will only draw the solvent atoms inside the kernel-radius of polymer atoms used for creating the density volume.

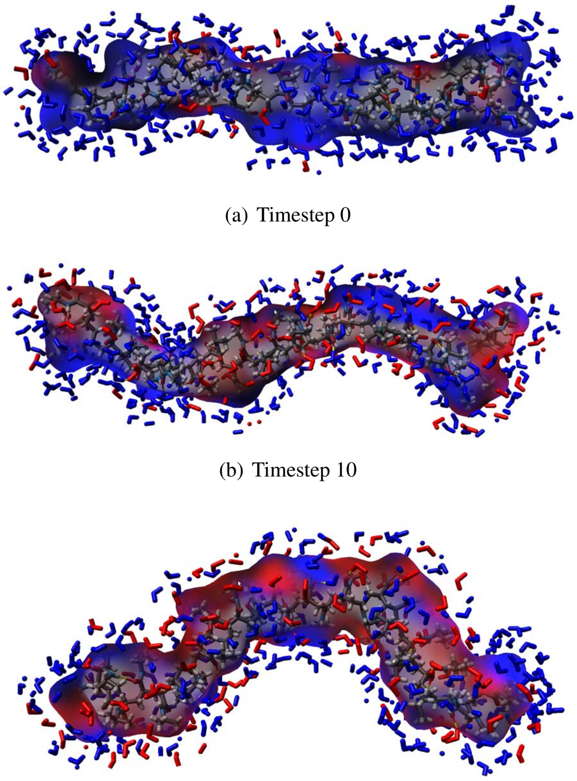
A similar approach is used for counting how many molecules of each solvent type are located in the hydration shell. All atoms are sent to the graphics card with additional information about the solvent type of the molecule they belong to. A framebuffer object which size equals the number of molecules is used as render target. For each atom, a vertex shader program is used to determine whether it is visible or not based on the density value of the 3D-texture as explained above. If the atom is visible, the pixel corresponding to the atom's molecule is set to the solvent type index. In the second step, all pixels of the resulting framebuffer texture are sent to the graphics card as points. This time, a framebuffer object which size corresponds to the number of solvent molecule types is used. For each visible atom, the value of the pixel corresponding to the stored solvent type is increased by one. This is accomplished using an additive blending function. Thereby, we count the number of molecules of each solvent type with at least one atom inside the polymer hydration shell.

5. Results

5.1. Evaluation by Domain Experts

For a first application of the new visualization methods, the trajectories of atomistic molecular dynamics simulations of PNIPAAm polymer strands using explicit solvent were used. The polymer strands with 30 monomers were simulated in water/methanol mixtures with different methanol concentrations.

Screenshots taken from these visualizations can be seen in all figures in this paper. In all images the filtering of the solvent is active, i.e. only the solvent molecules in the hydration shell around the polymer are shown and colored accordingly to a user-defined property. In Figure 6 methanol is colored red and water is colored blue. Further, the hydrogen bonds between the solvent and the polymer and between the polymer and the polymer are shown as red and yellow sticks (see Figure 2). Comparing the visualizations at different methanol concentrations over the simulation time, the following can be observed: The hydrogen bonds between water and the polymer are more stable than the ones between methanol and water. Furthermore, the hydrogen bonds in the polymer are only a few in the mixture and nearly none in the pure solvents. With the statistics for the hydrogen bonds, it can be seen that the hydrogen bonds between the solvent and the oxygen of the polymer are preferred to those between the solvent and the hydrogen in the amide group.


(a) Timestep 0

(b) Timestep 10

(c) Timestep 20

Figure 6: *Development of solvent concentrations over time. The increasing methanol concentration around the polymer (red) yields that methanol displaces water (blue) as solvent in our dataset. The solvent in the dataset consists of 90 mol% water and only 10 mol% methanol.*

With the new method for the visualization of the local solvent concentration it is even possible to see and compare the composition of the solvent around the polymer. This comparison can be seen in Figure 6. Here, the solvent around the polymer for a methanol concentration of 10 mol% in the water/methanol mixture is shown at the initial configuration (Figure 6(a)) and later in the simulation (Figure 6(b) & 6(c)). Looking at the solvent and the color of the surface around the polymer, it is evident that the methanol concentration

Table 1: *Performance measurements for various data sets in animation mode. HB denotes the time for calculating the hydrogen bonds for a timestep in the trajectory. #s-atoms denotes the number of solvent atoms and #p-atoms the polymer atoms accordingly.*

Data set	#s-atoms	#p-atoms	fps	HB
traj_30	38,214	575	22 fps	1.0 ms
xm0.1/traj	44,046	575	20 fps	1.2 ms
traj_50	67,725	955	15 fps	1.9 ms
traj_75	110,961	1,430	9 fps	3.5 ms
traj_diamant	168.204	4,822	6 fps	6.5 ms

around the polymer in equilibrium is much higher than in the bulk phase of the solvent represented by the first image in Figure 6. These results help domain experts to better understand the processes that take place in the PNIPAAm-water-methanol system.

5.2. Performance

All performance measurements were executed on an Intel Core i7-2600 CPU (4 × 3.4 GHz) with 8 GB RAM and an NVIDIA Geforce GTX560 (1 GB VRAM) at a resolution of 1024 × 768 pixels. For the 3D-volume texture we use for the polymer iso-surface, we chose a size of 128^3 voxels. This results in a good balance between accuracy and volume generation performance, because a volume size of 256^3 cuts the frame rate by a factor of 4 while the visual difference is barely noticeable. All frame rates have been taken in animation mode where the trajectory is interpolated for each frame. In this mode, the 3D-volume texture is recomputed for each frame. Atom movements are smoothed by linear interpolation between the discreet positions obtained from the trajectory.

Table 1 shows different data sets with their atom counts for the solvent and polymer parts. The frame rate (fps) is for animation mode. The calculation of the hydrogen bonds for a single timestep in the trajectory is shown in the column HB of Table 1. The measurements show that interactive frame rates are possible for data sets with over 100,000 atoms on standard hardware.

6. Conclusions and Future Work

With the novel visual analysis methods presented in this paper, domain experts can get better insight into the results of polymer-solvent molecular dynamics simulations. Even though the analysis results of our project partners given in Section 5.1 are still preliminary, they provided new insight in the simulation data. This is a promising first step in answering open questions in current research topics by using our visual analysis methods. The core of these techniques is to facilitate a better understanding of the structure and behavior of solvent around a solute. Therefore, they are specifically tailored to investigate the solvent structure around polymers, which are of special interest to our project partners. However, the methods can also be applied to similar problems such as protein-solvent systems.

Future work will focus on providing additional parameters for the analysis, such as polymer-solvent interactions or the kinetic energy, which are of great interest as well. This could help our project partners to further investigate the processes that lead to the conformation transition of large molecules like polymers, hydrogels and proteins in different solvents and mixtures.

Acknowledgments

This work was partially funded by the Germany Research Foundation (DFG) as part of the Collaborative Research Centre SFB 716. The computer simulations were performed on the supercomputer HP XC4000 at the Steinbuch Centre for Computing in Karlsruhe (Germany) under the grant LAMO. This work was partially carried out under the auspices of the Boltzmann-Zuse Society for Computational Molecular Engineering (BZS).

References

[BGB*08] Bidmon K., Grottel S., Bös F., Pleiss J., Ertl T.: Visual Abstractions of Solvent Pathlines near Protein Cavities. *Computer Graphics Forum (Proceedings of EUROVIS 2008) 27*, 3 (2008), 935 – 942. 3

[Con83] Connolly M. L.: Analytical molecular surface calculation. *J. Appl. Cryst. 16* (1983), 548–558. 2

[CSW06] Cheng H., Shen L., Wu C.: LLS and FTIR studies on the hysteresis in association and dissociation of poly (N-isopropylacrylamide) chains in water. *Macromolecules 39* (2006), 2325–2329. 2

[DBG10] Dias S., Bora K., Gomes A.: CUDA-based Triangulations of Convolution Molecular Surfaces. In *Proceedings of ACM International Symposium on High Performance Distributed Computing (HPDC '10)* (2010), pp. 531–540. 3

[DeL02] DeLano W. L.: *The PyMOL Molecular Graphics System*. DeLano Scientific, Palo Alto, CA, USA, 2002. http://www.pymol.org. 2

[dTLP07] de Toledo R., Lévy B., Paul J.-C.: Iterative methods for visualization of implicit surfaces on GPU. In *ISVC, International Symposium on Visual Computing* (Nov 2007), pp. 598–609. 2

[ER05] El-Rehim H. A. A.: Swelling of radiation crosslinked acrylamide-based microgels and their potential applications. *Radiation Physics and Chemistry 74* (2005), 111–117. 1

[FGE10] Falk M., Grottel S., Ertl T.: Interactive Image-Space Volume Visualization for Dynamic Particle Simulations. In *Proceedings of The Annual SIGRAD Conference* (2010), Linköping University Electronic Press, pp. 35–43. 3, 5

[GM10] Giard J., Macq B.: Molecular Surface Mesh Generation by Filtering Electron Density Map. *Journal of Biomedical Imaging 2010* (2010), 10:9–10:9. 3

[GRE09] GROTTEL S., REINA G., ERTL T.: Optimized Data Transfer for Time-dependent, GPU-based Glyphs. In *Proceedings of IEEE Pacific Visualization Symposium 2009* (2009), pp. 65–72. 2

[Gum03] GUMHOLD S.: Splatting illuminated ellipsoids with depth correction. In *Proceedings of VMV* (2003), pp. 245 – 252. 2

[HDS96] HUMPHREY W., DALKE A., SCHULTEN K.: VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics 14* (1996), 33–38. 2

[HLSR08] HADWIGER M., LJUNG P., SALAMA C. R., ROPINSKI T.: Advanced illumination techniques for GPU volume raycasting. In *ACM SIGGRAPH Asia 2008 courses* (2008), pp. 1–166. 3, 5

[HXM06] HÜTHER A., XU X., MAURER G.: Swelling of N-isopropyl acrylamide hydrogels in aqueous solutions of sodium chloride. *Fluid Phase Equilibria 240* (2006), 186–196. 1

[KBE09] KRONE M., BIDMON K., ERTL T.: Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1391–1398. 2

[KC05] KOLB A., CUNTZ N.: Dynamic Particle Coupling for GPU-based Fluid Simulation. In *Proceedings of Symposium on Simulation Technique (ASIM)* (2005), pp. 722–727. 5

[KE04] KLEIN T., ERTL T.: Illustrating magnetic field lines using a discrete particle model. In *Proceedings of VMV* (2004), pp. 387–394. 2

[KLR04] KOLB A., LATTA L., REZK-SALAMA C.: Hardware-based Simulation and Collision Detection for Large Particle Systems. In *ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware* (2004), pp. 123–131. 3

[LB06] LOOP C., BLINN J.: Real-time GPU rendering of piecewise algebraic surfaces. *ACM Trans. Graph. 25*, 3 (2006), 664–670. 2

[LBPH10] LINDOW N., BAUM D., PROHASKA S., HEGE H.-C.: Accelerated visualization of dynamic molecular surfaces. *Computer Graphics Forum (Proceedings of EUROVIS 2010) 29* (2010), 943–952. 2

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of ACM SIGGRAPH Computer Graphics and Interactive Techniques* (1987), pp. 163–169. 3

[LVRH07] LAMPE O., VIOLA I., REUTER N., HAUSER H.: Two-Level Approach to Efficient Visualization of Protein Dynamics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1616–1623. 2

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2003* (2003), pp. 154–159. 3

[MSS*93] MUKAE K., SAKURAI M., SAWAMURA S., MAKINO K., KIM S. W., UEDA I., SHIRAHAMA K.: Swelling of poly(N-isopropylacrylamide) gels in water-alcohol (C1-C4) mixed solvents. *Journal of Physical Chemistry 97* (1993), 737–741. 2

[OS07] ONO Y., SHIKATA T.: Contrary hydration behavior of N-isopropylacrylamide to its polymer, P(NIPAm), with a lower critical solution temperature. *Journal of Physical Chemistry B 111* (2007), 1511–1513. 2

[PGH*04] PETTERSEN E. F., GODDARD T. D., HUANG C. C., COUCH G. S., GREENBLATT D. M., MENG E. C., FERRIN T. E.: UCSF Chimera–a visualization system for exploratory research and analysis. *J. Comput. Chem. 25*, 13 (Oct 2004), 1605–1612. 2

[PSI*09] PAVLYUCHENKO V. N., SOROCHINSKAYA O. V., IVANCHEV S. S., KHAIKIN S. Y., TROUNOV V. A., LEBEDEV V. T., SOSNOV E. A., GOFMAN I. V.: New silicone hydrogels based on interpenetrating polymer networks comprising polysiloxane and poly(vinyl alcohol) networks. *Polymers for Advanced Technologies 20* (2009), 367–377. 1

[RE05] REINA G., ERTL T.: Hardware-accelerated glyphs for mono- and dipoles in molecular dynamics visualization. In *EuroVis05: EG/IEEE Symposium on Visualization* (2005), pp. 177–182. 2

[Ric77] RICHARDS F. M.: Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering 6*, 1 (1977), 151–176. 2

[SMT91] SCHILD H. G., MUTHUKUMAR M., TIRRELL D. A.: Cononsolvency in mixed aqueous solutions of poly(N-isopropylacrylamide). *Macromolecules 24* (1991), 948–952. 2

[SSKE05] STEGMAIER S., STRENGERT M., KLEIN T., ERTL T.: A Simple and Flexible Volume Rendering Framework for Graphics-Hardware-based Raycasting. In *Proceedings of the International Workshop on Volume Graphics '05* (2005), pp. 187–195. 3

[SWBG06] SIGG C., WEYRICH T., BOTSCH M., GROSS M.: GPU-Based Ray-Casting of Quadratic Surfaces. In *Eurographics Symposium on Point-Based Graphics* (2006), pp. 59–65. 2

[TKKW09] TANAKA F., KOGA T., KOJIMA H., WINNIK F. M.: Temperature- and tension-induced coil-globule transition of poly(N-isopropylacrylamide) chains in water and mixed solvent of water/methanol. *Macromolecules 42* (2009), 1321–1330. 2

[WDVH10] WALTER J., DEUBLEIN S., VRABEC J., HASSE H.: *High Performance Computing in Science and Engineering '09*. Springer, 2010, ch. Chemistry, Development of Models for Large Molecules and Electrolytes in Solution for Process Engineering, pp. 165–176. 2

[WEVH10] WALTER J., ERMATCHKOV V., VRABEC J., HASSE H.: Molecular dynamics and experimental study of conformation change of poly(N-isopropylacrylamide)-hydrogels in water. *Fluid Phase Equilibria 296* (2010), 164–172. 2

[WHV90] WINNIK F. M., H.RINGSDORF, VENZMER J.: Communications to the editor - methanol-water as a co-nonsolvent system for poly(N-isopropylacrylamide). *Macromolecules 23* (1990), 2415–2416. 2