Clutter Reduction in Rendering of Particle (Atom) Trajectories with Adaptive Position Merging

Bidur Bohara[1] and Bijaya B. Karki[1-3]
[1]School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, U.S.A.
[2]Department of Geology and Geophysics, Louisiana State University, Baton Rouge, U.S.A.
[3]Center for Computation and Technology, Louisiana State University, Baton Rouge, U.S.A.

Keywords: Trajectory Rendering, Clutter Reduction, Position Merging, Atomistic (Molecular) Visualization

Abstract: Visualization of position-time series data from molecular dynamics simulations of a material has to render atomic trajectories, and relevant structural and dynamical information. Clutter/occlusion associated with overlapping trajectories becomes serious even for moderate data sizes. We present an adaptive hierarchical scheme for merging multiple positions along trajectories to significantly reduce the number of points/line segments used for rendering. Our approach finds positions lying within a space window (cut-off distance) from a reference position and merges them into a single position. The window is then moved in time order with merging performed at each successive location. All original positions are thus processed to a reduced number of new (merged) positions, which are further merged with the same or a different cut-off to obtain even fewer positions. This hierarchical merging may continue several levels deep. Moreover, merging can be performed subject to constraint of information, which is displayed (color-coded) along individual trajectories. Both the trajectory geometry and underlying atomic structure become increasingly visible after merging so the nature and extent of atomic arrangements and movements can be better assessed.

1 INTRODUCTION

Molecular dynamics (MD) simulations of real material systems produce massive amounts of time-varying positional data for constituent atoms or molecules (Allen and Tidesley, 1987); (Peng et al. 2011). Visualization of such correlated data allows us to gain insight into the structural and dynamical behaviour of the material system under consideration. Different approaches depending on the nature of the atomic systems studied have previously been used to visualize the scattered MD data (see the Section 2). Two approaches for a rapid navigation through the data are animation and trajectories (Bhattarai and Karki 2009); (Li, 2005); (Humphrey et al., 1996). Particle (atom) trajectories allow a complete representation of position-time series by rendering positions of all atoms at all-time steps as points or line segments, so full simulation information is contained in a single display. As shown in Figure 1 (left), the main problem is that trajectory rendering becomes too crowded because there are simply many trajectories that are long, distributed in 3D space, and overlapped with each other. Note that the trajectories shown are unfolded so they become continuous curves extending in the space both inside and outside the simulation supercell.

The challenge is how we can render the atomic trajectories with a minimum clutter while preserving the dynamical and structural information contained in the data. The degree of trajectory crowdedness is expected to increase with the size of the data, which depend on the number of atoms and the number of time steps. Though the clutter cannot be completely avoided, it may be reduced to an acceptable level via repetitive processing and analysis of the data. Deciding whether to accept or reject the processed trajectories may require a domain specific knowledge. Various factors such as the geometrical shape, spatial confinement, and other physical properties of the trajectory are relevant. Furthermore, the information about the geometric arrangement of constituent atoms (i.e., atomic structures) also needs to be rendered together with the trajectories.

In this paper, we report an improvement in rendering of the atomic trajectories from MD

Bohara B. and B. Karki B..
Clutter Reduction in Rendering of Particle (Atom) Trajectories with Adaptive Position Merging.
DOI: 10.5220/0004720102640271
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (GRAPP-2014), pages 264-271
ISBN: 978-989-758-002-4

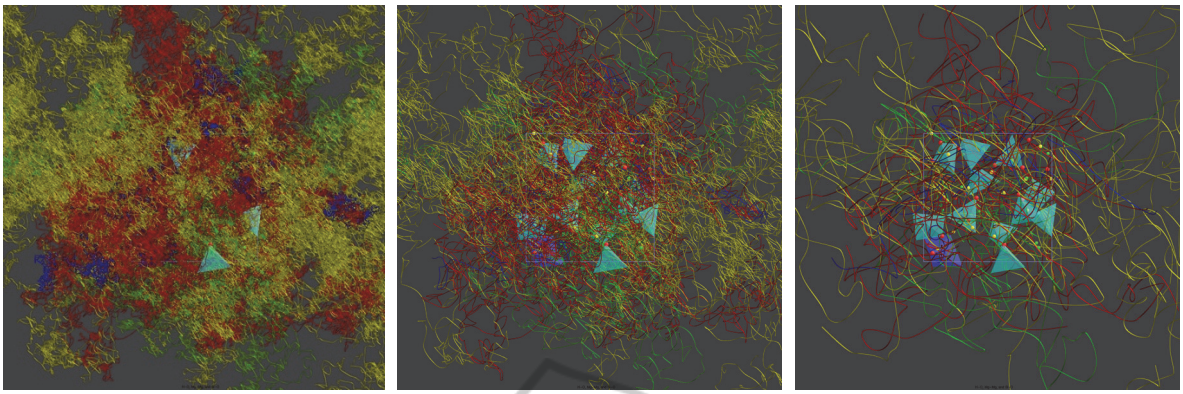

Figure 1: Visualization of molecular dynamics simulation data: atom trajectories and structure are rendered together. As multiple positions along the trajectories are merged with our adaptive hierarchical merging (described in section 3), the structure (consisting of Si-O polyhedra, H-O bonds, and floating Mg atoms) becomes increasingly visible (left to right).

simulations. We overcome the clutter/occlusion problem associated with the trajectories by position merging, encode useful information along the trajectories, and superimpose the atomic structure. While our approach should work for any MD data set, here we deal with atomic systems that are routinely simulated using the first-principles molecular dynamics (FPMD) method based on the fundamental equations of quantum mechanics. Two atomic systems considered are hydrous silicate ($MgSiO_3$+H_2O) and silica (SiO_2) liquids (e.g., Karki et al., 2010). The first system consists of 84 atoms in a supercell of length of 10.11Å and simulation run of 60,000 time steps (with each time step being 1 femtosecond). There are about 5 millions of data points in total. The second system consists of 72 atoms in a supercell of length of 10.32Å and simulation run of 1.2 million time steps. There are over 80 million data points in total. We consider these data sets to be of moderate size in the sense that the classical MD simulations based on pairwise potential functions can handle very large systems even exceeding one billion atoms (Nakano et al. 2007).

2 RELATED WORK

Atomistic (molecular) visualization is one of the most widely studied applications of computer graphics and visualization (Bhattarai and Karki 2009); (Li, 2005); (Humphrey et al., 1996); (Grottel et al., 2010); (Stone et al., 2011); (Zhang et al., 2007). To gain insight into MD and FPMD simulations, visualization has previously been exploited in various ways depending on the nature of the atomic systems studied. Few common examples include VMD (Humphrey et al., 1996), Molscript (Kraulis, 1991), XcrysDen (Kokalj, 1999), Atomeye (Li, 2003), etc. In addition to rendering the atomic configurations in various forms, they also support animation and trajectories. To the best of our knowledge, not much work was previously done in manipulating the atomic trajectories.

Trajectory rendering is also relevant for other movement-related data such as traffic network consisting of roads and trails (Buchin et al., 2011). Removing selected trajectories or portions of those trajectories, and line simplification of highly curved trajectories can help reduce the clutter to some extent (Borcan, 2012). It is relevant to point out that the pathline or trajectory technique is widely used in vector visualization (Shi et al., 2007) with a lot of efforts made on correct illumination and 3D rendering of lines (Zockler et al., 1996), and encoding the relevant information (Jones et al., 2007). In the case of MD simulations, the input position-time series directly represent the trajectory data so trajectory rendering displays the spatio-temporal behaviour of atomic system. Few ideas on constraining trajectory rendering in space and time have recently been explored (Bohara and Karki, 2012). Here, we present an improvement in atomistic visualization by using adaptive hierarchical (repeated) position merging based on spatial proximity for trajectory rendering and by superimposing instantaneous atomic structures.

3 POSITION MERGING ALONG TRAJECTORY

Constituent atoms of a simulated material system are expected to show non-uniform movements. This

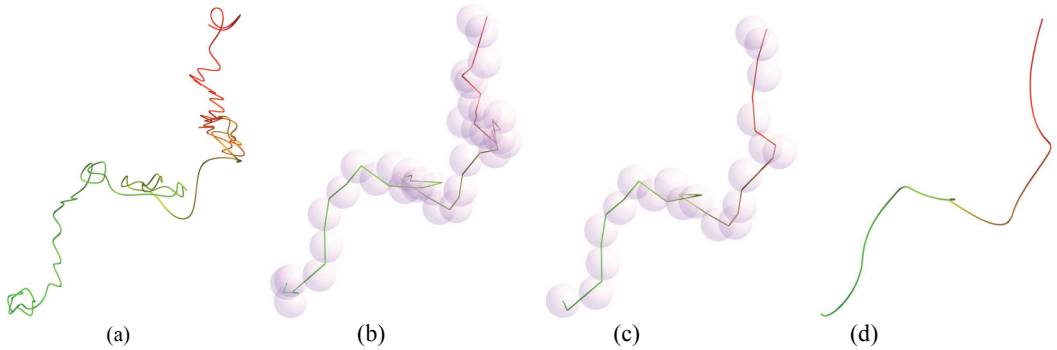

Figure 2: Space proximity based two-level position merging along trajectory. Spheres represent space windows (cutoffs). An original trajectory before merging (a), and the same trajectory after first merging (b), second merging (c) and smoothing (d). Time is encoded along the trajectory (green to red).

means that the atoms spend longer time in certain regions in 3D space than they do in other regions. Positions in slow regions where atoms are perhaps making oscillating (going back and forth) motions form short and crowded portions of the trajectories. A "proximity window" can be introduced based on the assumption of such confined atomic movements or localities so that positions lying in the window can be merged together. This window can be modeled as temporal proximity or spatial proximity. It is important that the 3D geometry of the trajectory be preserved during clutter reduction through merging of multiple positions. This also means that smaller amount of positional information is copied to the video memory for rendering.

3.1 Basic Merging Algorithm

We first consider the temporal proximity, which involves positions occupied by the concerned atom over a finite time interval. A time window is defined in terms of the number of successive MD steps, e.g., every 100 steps form a window. For each window, we find a representative point by calculating the mean (centroid) of all positions occupied at time steps belonging to that window. Single position is thus generated per window so the number of windows (N_{STEP}/N_{TW}) gives the number of positions to be rendered, where N_{STEP} is the total number of steps and N_{TW} is the size of the time window. Line segments are drawn between two successive mean positions to render the trajectories. The length of the line segments can vary significantly since the atoms move at different speeds at different time. Generally, N_{TW} needs to be very large in order to reduce the clutter in highly confined regions. This means that line segments in extended regions (where the atoms are moving fast without reversing directions) can be unusually long so the time-window based merging may not work well for the entire trajectory length.

Perhaps a more appealing approach is to apply spatial proximity, which considers positions located in the close vicinity of each other irrespective of time when these positions are occupied by the concerned particle. We define spatial proximity in terms of a finite space window (cut-off distance) so that all positions lying within such window are considered for merging. The idea is to scan all time steps to pick up positions whose distances measured from the reference position are shorter than the cut-off distance, and merge them together. The locality being approximated is thus a spherical region of radius equal to the cut-off distance (Figure 2).

Our position-merging algorithm for each trajectory works as follows:

Step 1: Set the space window at the starting (reference) position of an atom trajectory.

Step 2: Scan subsequent time steps to pick up positions lying inside the window until an outside position is encountered for the first time.

Step 3: Merge/map all positions found in the step 2 to single position.

Step 4: Start at the first position not yet picked up (new reference position), and do the steps 2 and 3.

Step 5: Move the window in the time order with merging performed at each successive location (repeating the step 4) until the end.

Thus, all positions are processed to generate a reduced set of new (merged) positions. If successive positions are well separated because either the cut-off distance is too small or atoms move too fast, merging is not effective at all and every window contains single but original position. On the other hand, if atoms have covered small distances because either atom are slowly moving/oscillating or the cut-off distance is too large, an arbitrarily large number of positions can be found in the window under

consideration to map to a single position. Note that processing atomic positions in the time order ensures the correct geometry of the trajectories while merging nearby positions as much as possible.

3.2 Adaptive Hierarchical Merging

After the first run of position merging using a cutoff distance, the clutter though it might have already reduced considerably is still not at an acceptable level and merged positions are still too close. So it may be desirable to apply additional merging with the same or larger cutoff. The next level merging then considers all new positions derived in the previous merging by placing a space window (of the same or bigger size) at the first merged position. Figure 2 illustrates two-level merging. As in the original merging, all positions lying within the current window at each successive location are picked up and merged. It is likely that the positions even after the second merging are still closely spaced, and/or again it may be desirable to apply an even larger cut-off. A third/higher level merging is then performed for the positions, which have undergone merging more than once. This adaptive hierarchical processing will continue until the number of repeatedly merged positions does not decrease anymore. This situation arises relatively fast if the cut-off distance is constant from the one level to the next. However, if the cut-off distance is increased progressively, the merging process may continue over many levels. The user has to specify when to stop either by not increasing the cut-off further or by visually assessing the crowdedness and correctness of the trajectories being increasingly approximated with multilevel position merging.

Merging is expected to be sensitive to information that we want to display (color code) along the trajectories. Examples of useful information include the distance travelled by atom, time spent, coordination states, bond events, merge count, mean squared error, etc. Such information is computed during merging process from original set of positions and appropriately mapped to the merged positions. Information extraction for coordination states involves the positions of nearby atoms. Atomic coordination ($C^i_{\alpha\beta}$) is the number of the nearest neighbour atoms of the same species (α) or other species (β) around a given atom of species α. Coordination is of multivariate type as it can take multiple values for a given window. Our approach to impose a coordination constraint while merging is that successive positions within a space window are merged only if they are in same coordination state. A position with different coordination state than the previous positions will start a new space window for merging next sequence of positions of the same coordination state. It is expected to result in more crowded trajectories than original merging.

3.3 Trajectory Rendering

To visualize the particle (atom) trajectories, we render a series of connected line segments for each trajectory by using the original positions (no merging) or new (merged) positions at different levels of merging. The user may choose to display single trajectory for a selected atom or a subset of trajectories corresponding to an atomic species or a group of atoms (e.g., SiO_4 unit), or trajectories for all constituent atoms.

Successive positions generated by the final hierarchical level merging are well separated (by more than the cut-off distance) so trajectories are no longer smooth and can be visually distracting. This becomes more serious when larger cut-offs are used. We perform a piece-wise smoothing of line segments by generating cubic and quadratic Bézier curves. A subset of merged positions along the trajectory is used as control points in the Bézier equation to generate the local curves. Additional information about the material system in question can be extracted on the fly and encoded along the trajectories using a color map.

4 RESULTS AND DISCUSSION

How much the trajectories are cluttered/crowded depends on the size of the data and the nature of the atomic movements. Here, we present our case studies by visualizing complete position-time series of two material systems (silicate and silica liquids) mentioned earlier.

4.1 Effects of Space Window Size

For a given data set, the size of space window (cut-off) applied controls the degree of clutter reduction to a great extent. Larger the cut-off, more positions merged along the trajectories and less crowdedness (Figure 3). However, too wide space windows can cover multiple localities and new positions produced by merging process may be too few to approximate the trajectory geometry to any acceptable level (Figure 3, right). There is a trade-off between cut-off (clutter measure) and the trajectory geometry.

The choice of the space cut-off requires domain

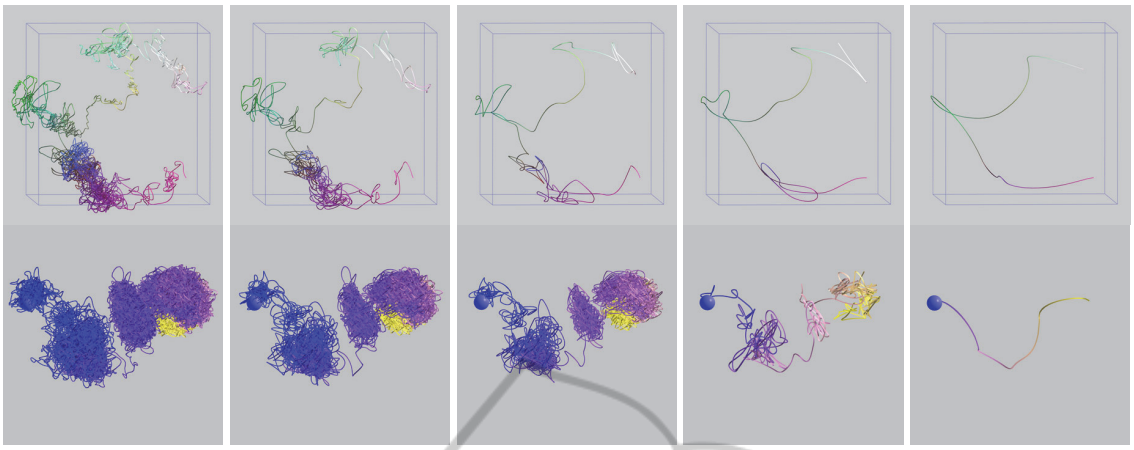

Figure 3: Effects of the space window (cutoff) size on the trajectory geometry. One H trajectory (top row) and one Si trajectory (bottom row) are processed using four different cutoffs. On increasing cutoff from left to right, the trajectories become simpler with local features disappearing more and more. The final trajectory (right) in each case fails to preserve the overall geometry. The color encodes time information along the original and processed trajectories.

specific knowledge and some pre-processing of a given position-time series data (such as radial distribution function and mean square displacement). Our adaptive approach allows the user to adjust the space cut-off by considering a few factors such as the number of trajectories, the shape and extent of trajectories, the desired level of clutter reduction, and the degree of trajectory approximation. For instance, if a single trajectory (one selected atom) or a few trajectories (atoms forming a structural unit or cluster) are visualized, a small cut-off may reduce the associated clutter to an acceptable level. However, if many (corresponding to all atoms of one species) or entire trajectories are rendered, relatively large cut-offs are needed.

4.2 Visualization of Trajectories

A detailed visualization of trajectories of hydrous silicate ($MgSiO_3$) and pure silica (SiO_2) liquids shows us that the atomic movements involve two types of motion: extended portions of the trajectories represent rapid continuous motion whereas the crowded portions represent confined (oscillating) motion. The atoms jump from one confined region to another in relatively short time interval.

For the hydrous system, each trajectory of each species (H, Mg, Si, and O) is obtained by rendering 60,000 successive positions occupied by the corresponding atom. Here we display 16 H trajectories obtained by processing about one million positions in the total (Figure 4). They start from the respective initial atomic positions all lying within the supercell and they eventually extend beyond the super-cell reaching as 3 or 4 times the supercell length. Note that H atoms are the most mobile species. It is difficult to trace these trajectories except in the outermost regions. Also, any atomic structures such as initial configurations shown by atomic spheres can hardly be seen. The H trajectories are approximated using our adaptive position merging approach with a cut-off 1.5Å and five levels of merging (Figure 4). This process suppresses local wiggling features and reduces the number of positions along the trajectories. Individual trajectories can be seen almost everywhere and all atomic spheres are visible. During visualization process, the user can explore the complete trajectory geometry with the aid of operations like rotation, translation, scaling, and highlighting.

The atoms in the silica liquid are very slow covering relatively small distances even over much longer simulation duration. In other words, the atoms are trapped locally most of the time. The Si trajectories obtained by rendering one million positions per atom are highly confined to narrow regions and they extend very little outside the supercell (Figure 5). Different trajectories tend to form distinct regions displayed with different colors but these regions overlap among each other. It is difficult to assess the shape and extent of these trajectories because the geometric structures are almost invisible. Our adaptive hierarchical position merging approach is expected to be useful.

4.3 Performance Analysis

The effectiveness of position merging along the trajectories can be assessed by counting the number

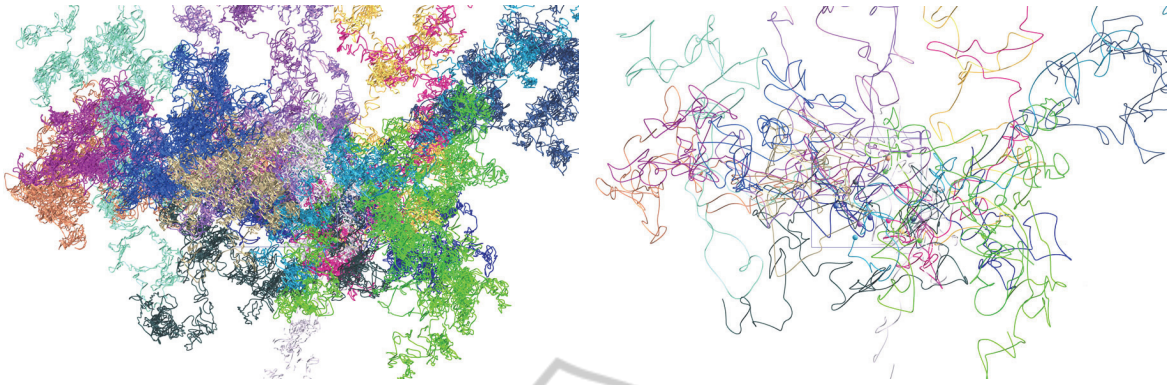

Figure 4: Trajectories of 16 H atoms (each colored differently) in hydrous silicate liquid after hierarchical merging.

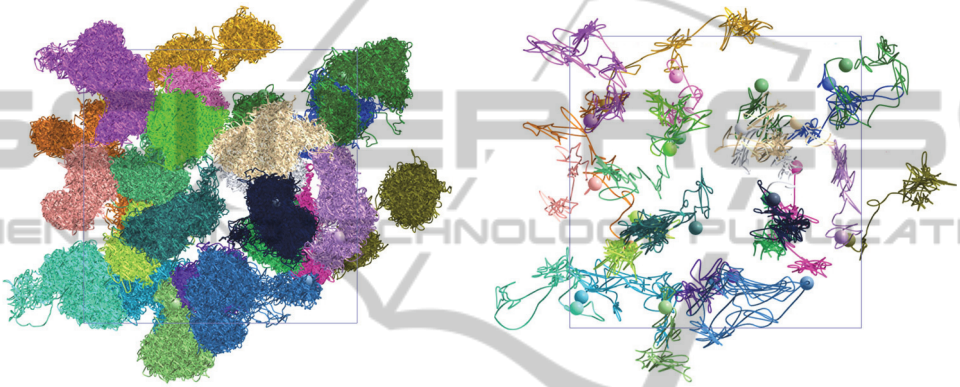

Figure 5: Trajectories of 24 Si atoms (each colored differently) in silica liquid after hierarchical merging.

of final (merged) positions used for trajectory rendering, examining the geometries (shape and extent) of individual trajectories, and having a measure (qualitative) of clutter reduction in visualization of trajectories and atomic structures. Key factors relevant for this assessment include space window size (cut-off distance), hierarchical merging level, and also imposed information constraint. Here, we present a performance analysis considering two cases: weak merging (with a narrow space window) and strong merging (with a wide space window) for each species trajectory for both material systems.

As shown in Table 1, the effects of window size (or cut-off) are substantial. For H trajectories, merging with a cut-off of 0.8Å involves 5 hierarchical levels and yields 16,450 final points compared to about 1 million original positions. The visibility of the trajectories improves considerably and some atomic spheres start to appear. When a larger cut-off of 1.5Å is used (again with 5 hierarchical levels) the number of final positions drops to 3,170 – a reduction by a factor of 300 with respect the original positions. The individual trajectories and underlying atomic structure become visible. For O trajectories, the number of final positions drops from about 9,000 (weak merging) to 1,770 (strong merging), compared to 2.6 million original positions. Similarly, the number of final positions drops by a factor of 4 and 6 between the two cases of merging for Mg and Si trajectories, respectively. A relatively larger effect is seen for Si trajectories because most of them are more confined.

Position merging along atomic trajectories for silica liquid is much more sensitive to window size than that for hydrous silicate liquid. A weak merging along Si trajectories using a cut-off of 0.5Å saturates at the 8^{th} level and generates around 57 thousands positions, compared to the original 24 million positions. The number of reduced (final) positions drops to around 4 thousands with somewhat larger cut-off (0.7Å). The overall reduction in the number of positions needed for trajectory rendering is by a factor of about 6,200. Thus, a much smaller set of processed positions, which represents only 0.016 % of the original positional data are rendered thereby dramatically reducing clutter in rendering of Si trajectories. Since O atoms cover longer distances, we use larger cut-offs of 0.8Å (weak merging) and 1.2Å (strong merging). The weak merging yields

Table 1: Number of atomic positions, original and after merging, of individual atom species in hydrous silicate liquid (top part), and silica liquid (bottom part) for weak merging (small cut-offs) and strong merging (large cut-offs). Number of positions after first level merging at corresponding cut-off is presented inside the parentheses. The last column represents the number of positions for a constrained merging with large cut-off.

	Original	Weak Merging			Strong Merging			
		Cutoff	Levels	Positions	Cutoff	Levels	Positions	Constrained merging
H	960,000	0.8	5	16,450 (24,710)	1.5	5	3,170 (7,160)	15,630 (16,830)
Mg	720,000	0.5	4	6,818 (7,975)	1.0	4	1,585 (2,520)	14,000 (14.015)
Si	720,000	0.5	4	2,920 (4,540)	1.0	3	510 (740)	1,975 (2.115)
O	2,640,000	0.8	5	8,945 (14,125)	1.5	4	1,770 (2,615)	5,480 (6,170)
Si	24 millions	0.5	8	57,420 (507,490)	0.7	5	3,865 (12,055)	51,530 (75,110)
O	48 millions	0.8	8	93,880 (989,570)	1.2	6	3,150 (12,290)	-

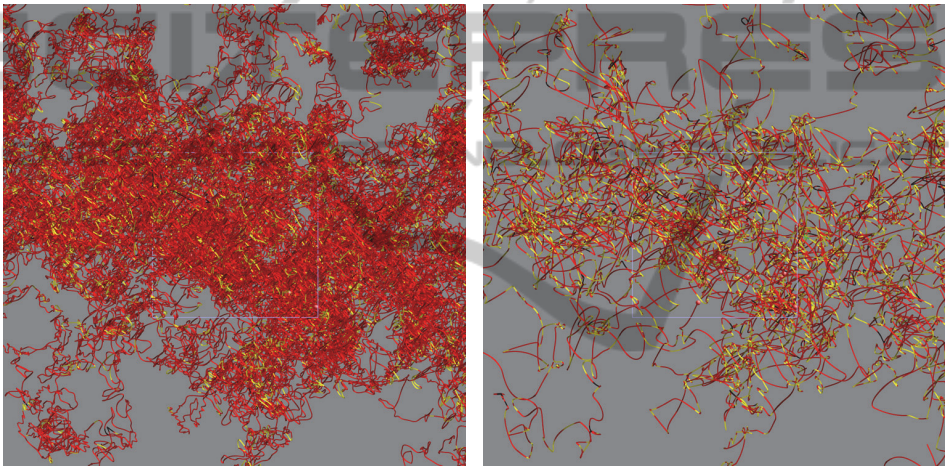

Figure 6: Trajectories of H atoms with color-coded coordination information with respect to O atom (red: singly coordinated, yellow double coordinated), before (left) and after (right) constrained merging with cut-off of 1.50Å.

little less than one hundred thousand positions, and the strong merging case yields much fewer positions (3,000 positions) for reduced trajectory, which represent about 0.0066 % of the original positional data and results in a much improved visibility.

The number of hierarchical levels needed to saturate merging varies somewhat depending on the space cut-off and atomic species. This number spans the range of 3 to 5 for two cases (weak and strong merging) considered for all species of hydrous silicate. The number is higher (5 to 8 levels) for Si and O trajectories of silica. Hierarchical merging tends to continue deeper for the cases of highly confined trajectories. The number of new (merged) positions along trajectories decreases as merging goes deeper the merger hierarchy. As shown in Table 1, the numbers of positions differ significantly between the first level and the final level of hierarchical merging in both weak and strong merging.

Finally, we consider the sensitivity of position merging to a coordination information constraint (Figure 6). As shown in Table 1, fewer levels are needed to saturate merging in the hierarchy, and the number of final positions generated by merging along H trajectories, with H-O coordination as constraint, increases from around 3,000 (for original merging) to above 15,000 for a cut-off of 1.5Å. Similar increases in the number of final positions were found for other cases.

5 CONCLUSIONS

We have presented an improvement in atomistic visualization of position-time series data obtained from molecular dynamics simulations of real

materials by rendering both the atom (particle) trajectories and instantaneous structures. To reduce the clutter caused by the crowded trajectories, we perform adaptive hierarchical merging of multiple positions along the trajectories. According to our approach, the multiple positions to be merged at each level are picked up using a proximity window, which is defined in terms of space window (distance cutoff). Our analysis shows that the number of effective positions needed to render the trajectories decreases dramatically under merging (with/out information constraint) and the processed (reduced) trajectories show significantly reduced clutter. We can further enhance the visualization process by encoding additional information (time, 3D position, coordination number, and merge count) along the trajectories. Improved trajectories allow us to better assess the nature and extent of the corresponding atomic movements. In particular, they suggest that atoms move via discrete jumps (hopping-like motion) in addition to continuous forward motion. More importantly, the underlying atomic structures become visible with all trajectories rendered. While moderate-size data sets containing several millions of data points (atomic positions) were considered in this study, we anticipate to extend the proposed position merging to larger data sets produced by large-scale molecular dynamics simulations.

ACKNOWLEDGEMENTS

This work is supported in part by a grant from National Science Foundation (EAR 1118869).

REFERENCES

Allen M. P., Tidesley D. J., 1987. *Computer Simulation of Liquids*, Oxford University Press.

Bhattarai D., Karki B. B., 2009. Atomistic visualization: Space-time multiresolution integration of data analysis and rendering. *Journal of Molecular Graphics and Modelling,* 27: 951-68.

Bohara B., Karki B. B., Rendering particle trajectories with color-coded information for atomistic visualization, *12th IASTED Int'l Conf. on Visualization, Imaging and Image Processing* (*VIIP'12*), 2012, 782-049: 35-42.

Borcan M., 2012. Improving Visualization of Trajectories by Dataset Reduction and Line Simplification. *Master Thesis 2012, Utrecht University*.

Buchin K., Demšar U., Slingsby A., Willems E., 2011. *Results of the break-out group: Visualization*, *Dagstuhl SeminarProceedings*.

Grottel S., Reina G., Dachsbacher C., Ertl T., 2010. Coherent culling and shading for large molecular dynamics visualization, *Eruographics/IEEE-VGTC Symp. on Visualization,* 29, 3.

Humphrey W., Dalke A., Schulten K., 1996. VMD – Visual molecular dynamics, *Journal of Molecular Graphics* 14 (1996) 33-38.

Jones C., Ma K.L., Ethier S., Lee W.L., 2007. An integrated visual exploration approach to particle data analysis. *Tech. Rep. CSE-2007-20, University of California at Davis*.

Karki B.B., Bhattarai D., Mookherjee M., Stixrude L., 2010. Visualization-based analysis of structural and dynamical properties of simulated hydrous silicate melt. *Physics and* Chemistry of Minerals 37:103-117.

Kokalj A., 1999. XCrySDen – a new program for displaying crystalline structures and electron densities, *Journal of Molecular Graphics and Modelling* 17 (1999) 176-179.

Kraulis P.J., 1991. MOLSCRIPT: A program to produce both detailed and schematic plots of protein structures, *Journal of Applied Crystallography* 24 (1991) 946-950.

Li J., 2003. Atomeye: an efficient atomistic configuration viewer, *Modelling and Simulation in Materials Science and Engineering 11* (2003) 173-177.

Li J., 2005. Atomistic visualization, in: S. Yip (Ed), *Handbook Materials Modeling,* Springer, pp. 1051-1068.

Nakano A., Kalia R. K., Nomura K., Sharma A., Vashishta P., Shimojo F., Duin A. C. T., Goddard W. A., Biswas R., Srivastava D.,2007. A divide-and-conquer/cellular-decomposition framework for million-to-billion atom simulations of chemical reactions, *Computational Materials Science,* 38:642-652.

Peng L., Kunaseth M., Durson H., Nomura K., Wang R., Kalia R. K., Nakano A., Vashishta P., 2011. Exploiting hierarchical parallelisms for molecular dynamics simulation on multicore clusters, *Journal of Supercomputing* 57: 20-33.

Shi K., Theisel H., Hauser H., Weinkauf T., Matkovic K., Hege H. C., Seidel H.P., 2007. Path line attributes – an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. *Topology-Based Methods in Visualization.*

Stone E.J., Vandivort K.L., Schulten K., 2011. Immersive out of core visualization of large-size and long-timescale molecular dynamics trajectories. *Lecture Notes in Computer Science,* 6939:1-12.

Zhang C., Callaghan S., Jordan T., Kalia R.K., Nakano A., Vashishta P., 2007. Paraviz: a spatially decomposed parallel visualization algorithm using hierarchical visibility ordering. *International Journal of Computational Science* 1: 407.

Zockler M., Stalling D., Hege H.C., 1996. Interactive visualization of 3D-vector fields using illuminated streamlines. *Proc. IEEE Visualization'96, San Francisco,* pp. 107-113.