Web-based Visualization and Analysis of Atmospheric Nucleation Processes

Jinzhu Gao[1], Rui Ding[2], Yi Liu[3], Bin Chen[4], Xiaohui Cui[5], J. Ilja Siepman[2]

[1]University of the Pacific
[2]University of Minnesota
[3]Chinese Academy of Sciences
[4]Louisiana State University
[5]Oak Ridge National Laboratory
jgao@pacific.edu[1], dingrui@umich.edu[2], liuyi@ict.ac.cn[3]

Abstract

Nucleation phenomena play a pivotal role in many atmospheric and technological processes. However, understanding atmospheric nucleation processes has been difficult due to the lack of effective data exploration tools. In this paper, we present a web-based tool that allows remote users to mine the wealth of particle-based nucleation simulation data through web-based visualization and analysis. The simple yet effective web-based user interface is designed to support collaborative research in the area of atmospheric nucleation. The web-based tool also incorporates both visualization techniques and clustering analysis algorithms to support effective pattern recognition in the molecular self-organization of the clusters.

Keywords: *Web-based Interaction, Remote Handling, Scientific Visualization, Similarity Measures, Planetary Atmospheres*

1. Introduction

The nucleation processes involve the activated formation of liquid or solid nuclei (molecular clusters) from gas-phase species and play a pivotal role in many atmospheric and technological processes [22]. Understanding how particles or phases nucleate and grow in a multi-component mixture has important practical implications from climate to microemulsions, gas separations, and nano-materials. Although “the classical theory” for vapor nucleation was established nearly a century ago, nucleation pathways in the atmosphere remain largely unexplained, partially due to the lack of effective tools that support large-scale data exploration.

By transforming[1] simulation data into understandable figures, visualization has been widely used in many different areas including physics, chemistry, biology, geography, etc. However, due to vast amounts of data produced by a typical nucleation simulation, keeping multiple copies of data at different locations becomes impractical. In recent years, cyber technology has become increasingly popular as more and more tools have been developed for web use. Using standard HTML, JavaScript and Java applets, web browsers can easily access data distributed across the Internet [14]. As a result, the web has become an essential part in developing collaborative applications.

In this paper, we describe the development of a web-based tool, CT-IANP (Cyber Tool for Investigating Atmospheric Nucleation Processes), to facilitate effective remote visualization and analysis of atmospheric nucleation processes. Figure 1 illustrates the complete process from data generation to exploration. The web-based tool is deployed on the web server. Through a simple yet effective web-based user interface, a group of remote users can easily visualize and analyze the data generated from a particle-based nucleation simulation (using extremely efficient Monte Carlo sampling techniques [8, 17, 21, 19, 20] and accurate force fields [7]). Intelligent control mechanisms are used for remote file access, interactive visualization, as well as simultaneous data access. As the number of possible configurations to be analyzed can be too large for the pattern to be identified easily, it is essential to provide the support for pattern recognition. To achieve the goal, a flocking based clustering algorithm is first applied and the clustering results are then organized in folders for easy manipulation. Java 3D is used in our implementation due to its high performance and high level of abstraction.

The rest of the paper is organized as follows. In Section 2, we describe previous work on remote visualization and clustering algorithms. Section 3 gives an overview of our design. We present our solution to assist visual pattern recognition in Section 4. Finally, we summarize our contributions and future directions in Section 5.

2. Related Work

Remote visualization typically involves visualization servers for retrieving data and computing visualization results, with one or multiple clients participating in the rendering and display of the visualization results to the user. Among remote visualization techniques proposed over the years, we are particularly interested in those work that involve developing advanced middleware and frameworks for collaborative data analysis and visualization. Brodlie *et al.* [6] assessed a selection of visualization systems and frameworks for their use in a collaborative environment. Shalf and Bethel [25] examined the impact of the Grid on visualization and suggested the need for a simulation environment in Grid research. Grimstead *et al.* [11] presented a collaborative grid enabled visualization environment that supports automated resource discovery. The Grid Visualization Kernel (GVK) [13] was proposed to exploit the power of the grid to provide visualization services. Beynon *et al.* [4] developed a filter-stream programming framework for data-intensive applications. Ahrens *et al.* [1] presented an architectural approach to handle large-scale visualization problems with parallel data streaming. Luke and Hansen [16] proposed a general framework capable of supporting multiple scenarios to partition a remote visualization system. Boier-Martin [5] presented a framework that allows visual representations of information to be customized and mixed. Many Eyes [27], a public website for uploading data and creating interactive visualization, supports collaborative visualization at a large scale.

To effectively retrieve information from a large-scale dataset, clustering algorithms have been widely used. It typically involves dividing a set of objects into a specified number of clusters [2]. The goal is to find an inherent structure in the data and to expose this structure as a set of groups. The objects within each group should exhibit a large degree of similarity, while the similarity between different clusters should be minimized [3]. There are two major clustering techniques: *Hierarchical* and *Partitioning*. *Hierarchical* techniques produce a nested sequence of partitioning, with a single, all-inclusive cluster at the top and a set of clusters of individual points at the bottom. The time complexity of a hierarchical clustering technique is quadratic [26]. *Partitioning* clustering methods seek to partition a collection of objects into a set of non-overlapping groups, so as to maximize the evaluation value of clustering. The time complexity of a partitioning technique is almost linear, which makes it a better choice for many applications. The best-known partitioning algorithms are the *K-means* algorithm and its variants [12]. These simple and straightforward algorithms are based on the firm foundation of analysis of variances. They start with a random initial cluster center and keep reassigning each object to cluster centers based on the similarity between the object and the cluster center. The reassignment procedure continues until a convergence criterion is met (e.g., it runs for a fixed number of iterations or the cluster result does not change after a certain number of iterations). The main drawback of the K-means algorithm is that the clustering result is sensitive to the selection of the initial cluster centroid and may converge to the local optima. Another limitation of the K-means algorithm is that it generally requires a prior knowledge of the probable number of clusters for a data collection.

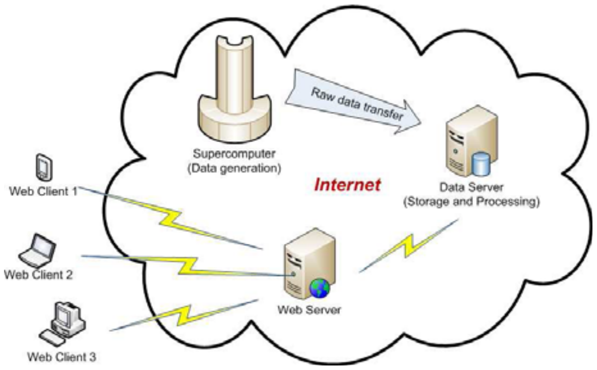
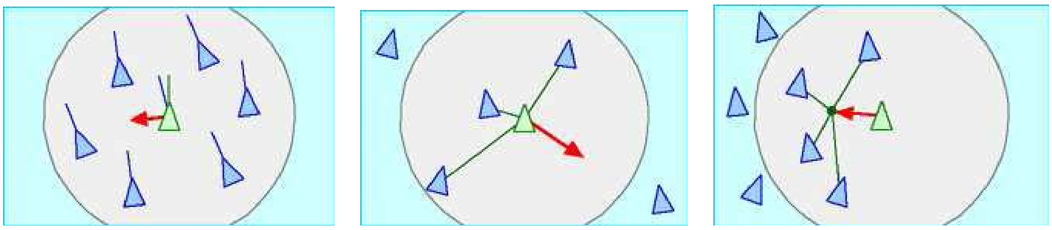


Figure 1. The complete process from data generation to exploration



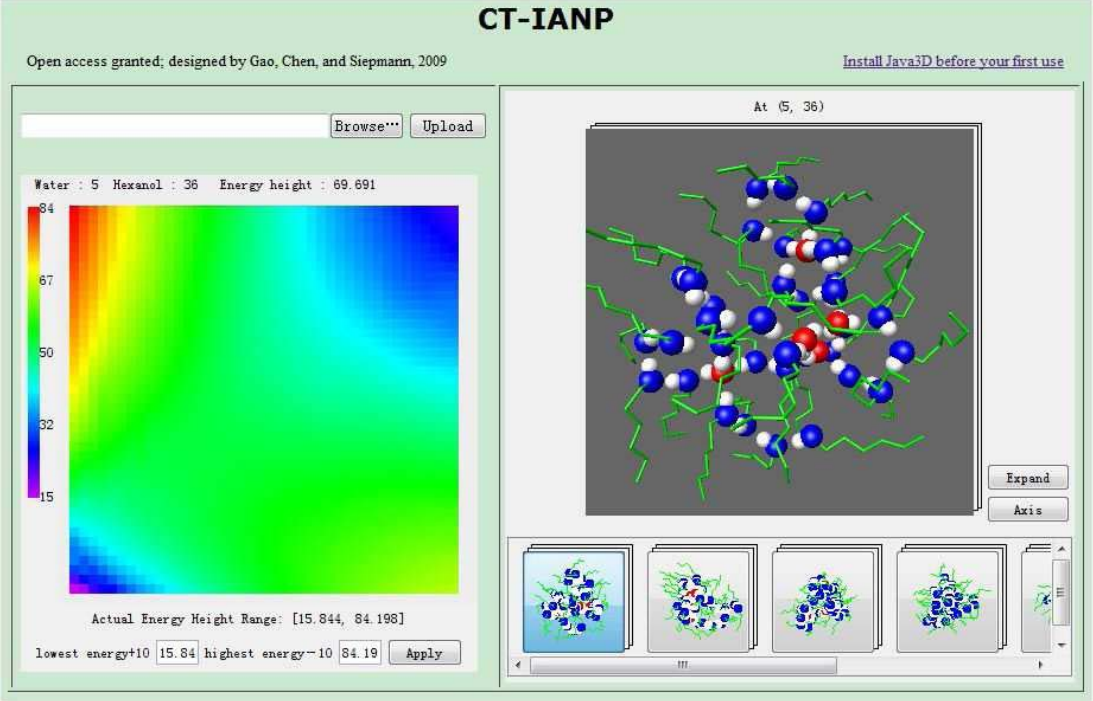
(a) (b) (c)

Figure 2. The three basic rules in the boid [24]: (a) Alignment: Steering toward the average heading and match the velocity of the neighbor flock mates. (b) Separation: Steering to avoid collision with other boids nearby. (c) Cohesion: Steering to the average position of the neighbor flock mates

To deal with the limitations that exists in the traditional partitioning methods, researchers have proposed several clustering approaches inspired by collective behaviors in biology, such as *Genetic Algorithm (GA)* [12], *Particle Swarm*

Optimization (PSO) [9], *Ant Clustering [15]*, and *Flocking-based approaches [18, 23]*. Among these algorithms, Ant clustering algorithms and flocking-based approaches do not require a prior knowledge of the probable number of clusters. Ant clustering algorithms were inspired by the clustering of corpses and eggs observed in the real ant colony. The drawback of these algorithms is that the objects do not have mobility themselves. The data objects movements have to be implemented through the movements of a small amount of ant agents, which slows down the clustering speed. Flocking [23] is a bio-inspired computational model for simulating the animation of a flock of entities called *boids*, as seen in bird flocks and fish schools. In this model, each boid makes its own decisions on its movement according to a small number of simple rules that react to the neighboring members in the flock and the environment it can sense. Three basic rules are demonstrated in Figure 2. These simple local rules of each boid eventually generate complex global behaviors of the entire flock [24]. Compared with ant clustering algorithms, flocking-based approaches need less iterations to generate an acceptable clustering result. Flocking behavior has already been used for time-varying data visualization by Moere and his colleagues [18]. In our implementation, we also chose the flocking-based clustering approach due to its efficiency and effectiveness. We believe it could give us a *global* view of a simulation data. However, to assist pattern recognition, we add a new rule that requires a flock agent should stay close to those agents with similar features and stay away from those agents with dissimilar features.

3. System Overview


CT-IANP
Open access granted; designed by Gao, Chen, and Siepmann, 2009
Install Java3D before your first use
Browse…
Upload
84
67
50
32
15
Expand
Axis
Apply

Water : 5 Hexanol : 36 Energy height : 69.691

At (5, 36)

Actual Energy Height Range: [15.844, 84.198]

lowest energy+10 15.84 highest energy−10 84.19

Figure 3. An example of the graphical user interface design

The web-based tool we describe in this paper is developed mainly for investigating atmospheric nucleation processes, although the design ideas should be applicable to other similar web-based tools. Our goal is to give remote users a collaborative workspace that promotes knowledge discovery by supporting interactive large-scale data exploration. As our targeted users are nucleation researchers, the tool is designed and developed in a way that users do not need to be an expert in computer science or networking or visualization in order to interactively analyze their data. That is, the tool must exhibit good usability. A simple yet effective user interface along with effective data visualization and analysis support allow people to use the tool by following three simple steps: (1) open a web browser; (2) type in the homepage URL, for example, http://[server's IP address]:8080/visualization/; (3) choose a dataset and then interact with the tool.

3.1. Simulation Data

A particle-based nucleation simulation typically produces two kinds of data sets. Table 1 gives an example of the data set for displaying the nucleation free energy surface and Table 2 describes the data set for displaying aggregate structures. In table 1, the three columns represent number of water molecules, number of hexanol molecules, and the corresponding nucleation free energy respectively. Table 2 provides the coordinates of each molecule (and its atoms).

3.2. User Interface Design

Table 1. Free energy height data. The size of our test data is 81 KB

Num of water molecules	Num of hexanol molecules	Free energy
0	1	21.577
0	2	30.622
1	0	17.286
. . .	. . .	. . .

Table 2. Aggregate structure data. The size of our test data is 673.291 MB

1 (1st configuration)		
19 (num of mocules)	0 (num of water)	19 (num of hexanol)
1 (1st molecules)	3 (hexanol)	8 (num of units)
219.49	-202.69	118.25
219.00	-201.67	119.12
219.39	-200.30	118.55
. . .	. . .	. . .

Figure 3 demonstrates the user interface design of the web-based tool. The 2D image on the left, created using the dataset in Table 1, shows the nucleation free energy contour map [8, 17, 21, 19, 20]. The color at each pixel on this map implies the magnitude of the nucleation free energy for the particular aggregate that this pixel refers to, which is controlled by a color scale bar. When the user moves the cursor over

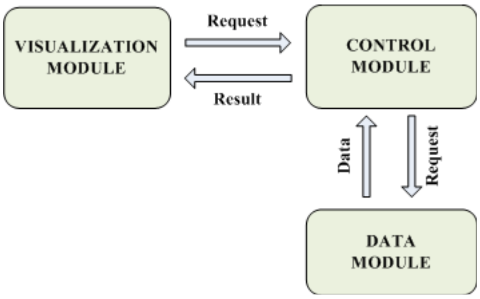


Figure 4. The relationship between three modules: Visualization Module, Data Module and Control Module

this interactive map, the top side bar shows the aggregate composition (i.e., the number of water and hexanol molecules) as specified by this point on the map and its nucleation free energy. On the right, molecular structures for all possible configurations stored in Table 2 for this aggregate are displayed. Web-based folders with graphic thumbnails demonstrate the clustering results for pattern recognition. The user could open a folder to view the configurations that have strong similarities in features such as molecular topology.

3.3. Module Design

The tool consists of three modules: (1) Visualization module, (2) Data module, and (3) Control module. The relationship between these modules is illustrated in Figure 4. The visualization module sends requests to the control module for the display content while the control module sends requests to the data module for the data that is required for satisfying the display request.

The visualization module is in charge of displaying visualization results on a webpage which consists of HTML, JSP and applets. Efficiency and scalability are the two major concerns in our implementation. Our design is based on two applets that are capable of communicating with each other at will. One applet handles the information about the raw data, including the description of the molecule types in the binary mixture, the current free energy range and its color scale bar, boxes to change the free energy range, a 2D color map of the free energy landscape, the location of the cursor specified in terms of the number of water and hexanol molecules, and the free energy for this aggregate size. The layout can be seen in Figure 3. After the raw data file is processed, a 2D image is shown in the main display panel while the range of the free energy height and the scale are updated accordingly. When a user moves the mouse over a particular point on the 2D image, the data, including the number of water molecules and the number of hexanol molecules, both encoded at the location of the mouse, is extracted and shown on the screen. The user can specify the range of the free energies of interest, which will also update the scale bar.

The other applet supports the 3D visualization of aggregate structures containing a specific number of water and hexanol molecules, as demonstrated in Figure 3. Red spheres represent *oxygen*, blue spheres represent *nitrogen*, white spheres represent *hydrogen*, and green sticks represent *carbon chains*. A user is able to perform operations such as Zoom in/out, translation, or rotation in the display panel.

The task of the data module is to parse the data file. We have two input data files: the free energy height data file and the aggregate structure data file. Since an aggregate structure data file may contain many thousands of configurations, the response to a user query for a particular aggregate in such a big file will definitely be very slow. Therefore, we must pre-process the data during the data uploading process, not only to reduce system response time, but also to make the system run more effectively. After pre-processing, the large file is partitioned into smaller files, indexed by the number of hexanol molecules and the number of water molecules. Uploading and parsing the data is a one-time process. In our experiments with a 673.291 MB dataset, the pre-processing took about 3 minutes on a web server with a 2.0 GHz AMD Turion(TM) 64 X2 Dual-Core processor and 3 GB RAM.

The control module is responsible for web-based scheduling, data forwarding as well as error handling. Through HTTP, it supports the communication between applets and the server. Each time an applet requests a dataset from the server, it first constructs a URL object and packages the type of the data in the object. Then, an HTTP connection is set up with the server. The data can now be transmitted between the applet and the server as an input string or an output string.

4. Visual Pattern Recognition

Traditionally, nucleation researchers have to scan through hundreds or even thousands of aggregate structures in order to identify patterns in the molecular self-organization of the clusters, which can be very time-consuming and ineffective. Therefore, to speed up the knowledge discovery process, it is essential for the web-based tool to provide support for effective pattern recognition. To achieve the goal, we provide mechanisms that support quick identification and visualization of groups of similar configurations. In this section, we first define the similarity metric we used in our implementation. Then we describe two ways to view clusters: flocking-based representation and web-based file folder representation. Flocking-based representation displays clustering results produced by a flocking-based clustering algorithm. It shows the relationships between all possible configurations for an aggregate. It gives the user a *global* and *continuous* view. Web-based file folder representation uses file folders to record groups of similar aggregate structures. It allows the user to analyze one group at a time and quickly identify subtle differences between similar aggregate structures. It gives the user a *local* and *discrete* view. Gaining knowledge from these two views, a nucleation researcher can quickly and easily go through all aggregate structures for possible patterns or other interesting features.

4.1. Similarity Metric

For a specific aggregate, there can be hundreds or thousands of possible configurations. Identifying groups of configurations with similar structures for a specific aggregate is important for nucleation researchers to understand their simulation data. One of our goals is to provide mechanisms that could automatically group similar aggregate structures. Before we provide any visual assistance, the similarity between any two aggregate structures needs to be defined and measured in a clustering analysis.

In our experiments with the particle-based nucleation simulation data, we aim to identify groups of configurations that have strong similarities in molecular topology.

Many features should be considered. For example, the centroid of an aggregate structure can be an important feature. When considering this feature, the similarity is defined by the Euclidean distance between the centroids of two aggregate structures. That is

$$D_{ij} = | G_i - G_j | \qquad (1)$$

The centroid of the i^{th} aggregate structure, G_i, is defined by

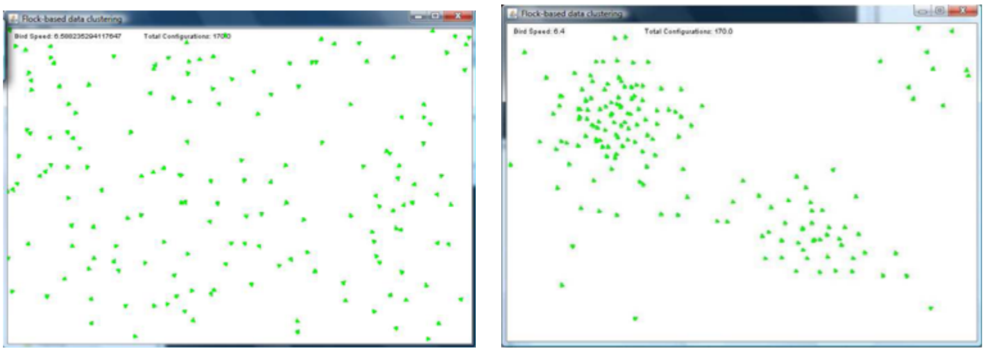
$$G_i = (\sum_{k=1}^{n_i} (w_k \times \overline{a}_k)) / n_i \qquad (2)$$

Where n_i is the number of atoms in the i^{th} aggregate structure, w_k is the weight of the k^{th} atom, and $\overline{a}_k$ is the magnitude of the vector (x_k, y_k, z_k). Here x_k, y_k and z_k define the location of the k^{th} atom.

If the distance, D_{ij}, is within a user-specified limit, ε, we consider these two structures similar when evaluating the *centroid* feature.

Different users might have different definition of similarity and consider a different set of features. For example, some users of our web-based tool might be interested in the symmetry of aggregate structures which can then be considered as another important feature. To serve users with various research interests, we provide a flexible interface for the users to choose their own similarity metric. The Vector Space Model (VSM) is used to facilitate our implementation [10]. The features of m aggregates can be represented as a set of vectors $X = \{x_1, x_2, \ldots, x_m\}$, where the vector x_i is called *feature vector* that contains values for features of the *ith* aggregate and it corresponds to a dot in the multi-dimensional space. To differentiate the importance of n features, another weight vector $w' = \{ w_1^{'}, w_2^{'}, \ldots w_n^{'} \}$ is also used where $w_i^{'}$ represents the significance of the i^{th} feature in the similarity evaluation. The similarity between the i^{th} aggregate and the j^{th} aggregate, $Sim(i, j)$, can then be calculated by the equation:

$$Sim(i, j) = (\sum_{k=1}^{n} (w_k^{'} \times | x_{i,k} - x_{j,k} |)) \qquad (3)$$


Flock-based data clustering

Bird Speed: 6.588235294117647 Total Configurations: 170.0

Flock-based data clustering

Bird Speed: 6.4 Total Configurations: 170.0

Figure 5. Spatial distribution of 170 configurations processed by the Flocking Clustering algorithm. (a) Initial distribution. (b) Distribution after 100 iterations

4.2. Flocking-based Representation

In our flocking-based representation, each configuration is represented as one bird agent in a two-dimensional virtual scene. Similar to a bird flock in the real world, the bird agents that share similar features will automatically group together as a bird-agent flock and other bird agents that have different features will keep away from this flock. An example is shown in Figure 5. The velocity of each agent A with position P_A depends on the agent X with position P_X in its neighborhood. A set of local behavior rules, including the alignment rule, the cohesion rule, and the separation rule, are applied to compute the velocity. The agent's velocity also depends on the feature similarity and dissimilarity compared to other agents. The mathematical definitions of the three behavior rules and the similarity rule are described below.

- **Alignment Rule:** The alignment rule, as shown in Figure 2(a), acts as the active flock bird tries to align its velocity vector with the average velocity vector of the flock in its local neighborhood. The degree of locality of this rule is determined by the sensor range of the active flock particle and is represented diagrammatically by the circle. The mathematical definition of this rule is:

$$|P_X - P_A| \leq d_1 \cap |P_X - P_A| \geq d_2 \Rightarrow v_{\mathrm{ar}} = \sum_X v_x / n \tag{4}$$

  Where v_{ar} is the velocity driven by the alignment rule, d_1 and d_2 are pre-defined distances and v_x is the velocity of the agent X.

- **Separation Rule:** The separation rule, as shown in Figure 2(b), acts as the active flock particle tries to pull away before crashing with another particle. The mathematical definition of this rule is:

$$|P_X - P_A| \leq d_1 \Rightarrow v_{sr} = \sum_X \frac{|P_X - P_A|}{||P_X - P_A||} \tag{5}$$

  Where v_{sr} is the velocity driven by the separation rule and d_1 is a pre-defined distance.

- **Cohesion Rule:** The cohesion rule, as shown in Figure 2(c), acts as the active flock particle tries to orient its velocity vector in the direction of the centroid (average spatial position) of the local flock.

$$||P_X - P_A|| \leq d_1 \cap ||P_X - P_A|| \geq d_2 \Rightarrow v_{cr} = \sum_X \frac{|P_X - P_A|}{||P_X - P_A||} \tag{6}$$

  Where v_{cr} is the velocity driven by the cohesion rule, d_1 and d_2 are pre-defined distances.

- **Feature Similarity and Dissimilarity Rule:** The flock agent tries to stay close to other agents that have similar features. In our experiments, the agents feature is represented by the location of the centroid of an aggregate structure. The strength of the attracting force is proportional to the distance between the agents and the similarity between the agents' features.

$$v_{ds} = \sum_X (W \times Sim(A, X) \times ||P_X - P_A||) \tag{7}$$

Where v_{ds} is the velocity driven by the feature similarity, W is a pre-defined constant, *Sim(A,X)* represents the similarity between the agent A and X.

The flock agent tries to stay away from other agents that are dissimilar based on the chosen feature. The strength of the repulsion force is inversely proportional to the distance between the agents and the similarity value between the agents' features.

$$v_{dd} = \sum_{X} \left(W \times \frac{1}{Sim(A,X) \times \| P_X - P_A \|}\right) \tag{8}$$

Where v_{dd} is the velocity driven by the feature similarity, W is a pre-defined constant.

To achieve comprehensive flocking behavior, the actions of all the rules are weighted and summed to give a net velocity vector for the active flock agent.

$$v = w_{sr} \cdot v_{sr} + w_{ar} \cdot \mathrm{v}_{ar} + w_{cr} \cdot v_{cr} + w_{ds} \cdot v_{ds} + w_{dd} \cdot v_{dd}$$

Where w_{sr}, w_{ar}, w_{cr}, w_{ds} and w_{dd} are weights.

By showing the relationships between all possible configurations for an aggregate, the flocking-based representation provides the user with a *global* and *continuous* view.

4.3. Web-based File Folder Representation

Guided by the similarity metric and user-defined criteria, we are able to partition aggregate structures into groups. Within each group, the distance between the centroids of any two structures should be smaller than ε. To represent groups effectively on a webpage, web-based file folders are developed to give the user a *local* and *discrete* view.

As described in Section 3, the data module needs to parse the large data file of aggregate structures and partition it into smaller files, one for each aggregate. To facilitate the display of grouping results, we add groups as a subdirectory in the file directory structure, as shown in Figure 6.

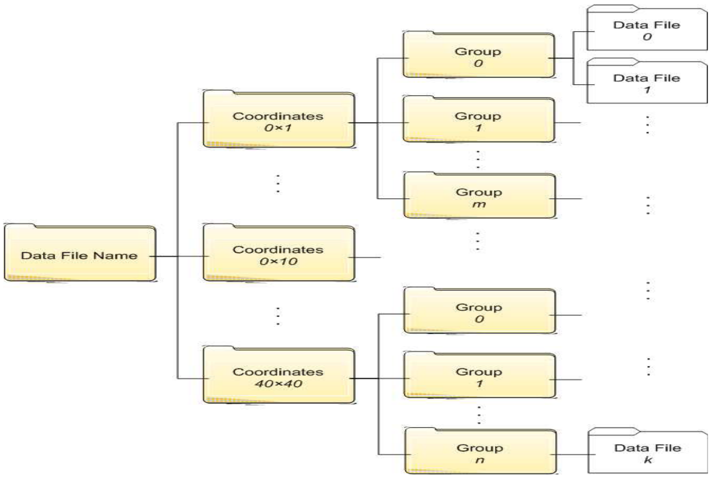


Figure 6. The file directory structure after data parsing

Of course, an effective web-based tool design also relies on the visual layout of the interface as we aim to "*make technology that conforms to people*". To achieve this goal, we provide both a thumbnail view and a detail view for each file folder and each individual molecule structure. The thumbnail view shows file folders or aggregate structures as graphic thumbnails, giving a preview of the visualization result. The detail view shows the visualization result in the original resolution. The user can easily switch between those two views. Under the detail view, the user is able to perform operations such as zooming in or out, translation, or rotation.

As demonstrated in Figure 7, we also provide two useful operations for the user to manipulate the visual representations of aggregate structures: *Expand* and *Axis*. Expand allows the user to open a folder to analyze the visualization results of all molecule structures in one group. Selecting the Axis option reveals the axis of the coordinate system, which highlights the relations between atoms.

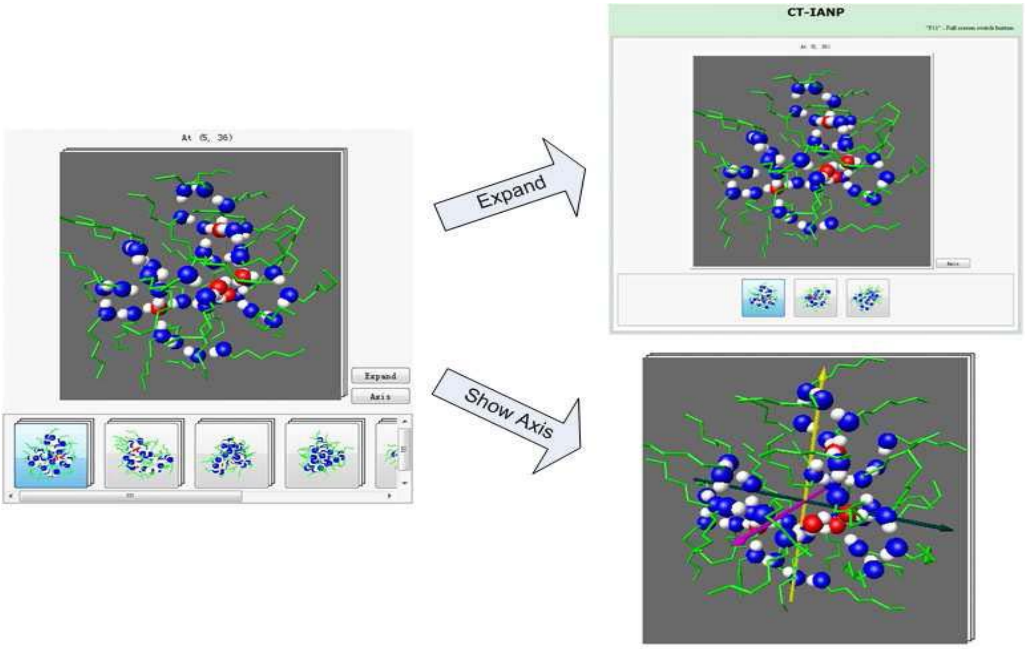


Figure 7. Demonstration of two possible operations: Expand and Axis. The image on the left shows file folders in either thumbnail view or detail view. When the Expand operation is applied, the visual representations of aggregate structures in the corresponding group are display, also in either thumbnail view or detail view

5. Conclusion And Future Work

In this paper, we describe the design and implementation of a web-based tool for visualizing and analyzing atmospheric nucleation processes. It allows researchers at geographically distributed locations to explore large-scale data simultaneously without keeping local copies. The tool also provides pattern recognition services that further speed up the knowledge discovery process. In the future, we would like to enhance the tool to provide more data analysis functionalities as well as more advanced similarity metrics. The user interface can also be improved to provide more effective services to enhance collaborative research experience.

Acknowledgments

This work was supported by the Minnesota Supercomputing Institute’s Seed Grant Program and the National Science Foundation (ITR-0428774, CBET-0756641, and CHE/MCB-0448918). The authors would like to thank Professor Elena Machkasova from the University of Minnesota at Morris, Samuel Keasler, Hyunmi Kim, and Ricky Nellas from the Louisiana State University at Baton Rouge, and the anonymous reviewers for their useful comments and suggestions.

References

[1] J. Ahrens, K. Brislawn, K. Martin, B. Geveci, C. C. Law, and M. Papka. Large-scale data visualization using parallel data streaming. IEEE Computer Graphics and Applications, 21(4):34–41, 2001.

[2] M. Anderberg. Cluster Analysis for Applications. Academic Press, Inc., New York, NY, 1973.

[3] P. Berkhin. Survey of clustering data mining techniques. Accrue Software Research Paper, 2002.

[4] M. D. Beynon, A. Sussman, T. Kurc, and J. Saltz. Optimizing execution of component-based applications using group instances. CCGrid, pages 56–65, 2001.

[5] I. M. Boier-Martin. Adaptive graphics. IEEE Computer Graphics and Applications, 23(1):6–10, 2003.

[6] K. Brodlie, D. Duce, J. Gallop, J. Walton, and J. Wood. Distributed and collaborative visualization. Computer Graphics Forum, 23(2):223–251, 2004.

[7] B. Chen, J. Potoff, and J. Siepmann. Monte carlo calculations for alcohols and their mixtures with alkanes. transferable potentials for phase equilibria. 5. united-atom description of primary, secondary and tertiary alcohols. J. Phys. Chem. B 105, pages 3093–3104, 2001.

[8] B. Chen, J. Siepmann, and M. Klein. Simulating vapor-liquid nucleation of water: A combined histogram-reweighting and aggregation-volumebias monte carlo investigation for fixed-charge and polarizable models. J. Phys. Chem A 109, pages 1137–1145, 2005.

[9] X. Cui and T. E. Potok. Document clustering using particle swarm optimization. In Proceedings of IEEE Swarm Intelligence Symposium 2005, 2005.

[10] B. Everitt. Cluster Analysis, 2nd Edition. Halsted Press, New York, 1980.

[11] I. J. Grimstead, N. J. Avis, and D. W. Walker. Automatic distribution of rendering workloads in a grid enabled collaborative visualization environment. In SC ’04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing, page 1, Washington, DC, USA, 2004. IEEE Computer Society.

[12] J. A. Hartigan. Clustering Algorithms. John Wiley and Sons, Inc., New York, NY, 1975.

[13] P. Heinzlreiter, D. Kranzlmuller, and J. Volkert. Network transportation and optimization for grid-enabled visualization techniques. Neural, Parallel Sci. Comput., 12(3):307–328, 2004.

[14] M. Jern. 3d data visualization on the web. In Proceedings of the 1998 Conference on MultiMedia Modeling, page 90, 1998.

[15] N. Labroche, N. Monmarch´e, and G. Venturini. Antclust: Ant clustering and web usage mining. In Proc. of Genetic and Evolutionary Computation Conference, pages 25–36, 2003.

[16] E. J. Luke and C. D. Hansen. Semotus visum: a flexible remote visualization framework. In Proc. of IEEE Visualization, pages 61–68, 2002.

[17] M. McKenzie and B. Chen. Unravelling the peculiar nucleation mechanisms for non-ideal binary mixtures with atomistic simulations. J. Phys. Chem. B 110, pages 3511–3516, 2006.

[18] A. V. Moere. Information flocking: Time-varying data visualization using boid behaviors. In Proceedings of the Eighth International Conference on Information Visualization, pages 409–414, 2004.

[19] R. Nellas, B. Chen, and J. Siepmann. Dumbbells and onions in ternary nucleation. Phys. Chem. Chem. Phys. 9, pages 2779–2781, 2007.

[20] R. Nellas, S. Keasler, and B. Chen. Molecular content and structure of aqueous organic nanodroplets from the vapor-liquid nucleation study of the water/n-nonane/1-alcohol series. J. Phys. Chem. A 112, pages 2930–2939, 2008. [21] R. Nellas, M. McKenzie, and B. Chen. Probing the nucleation mechanism for the binary n-nonane/1-alcohol series with atomistic simulations. J. Phys. Chem. B 110, pages 18619–18628, 2006.

[22] L. Ott and R. Finke. Supersensitivity of transition metal nanoparticle to initial precursor concentration and reaction temperature: Understanding its origins. Text Mining Workshop, KDD, 2000.

[23] C. Reynolds. Flocks, herds, and schools: a distributed behavioral model. Computer Graphics, 21(4):25–34, 1987.

[24] C. W. Reynolds. Steering behaviors for autonomous characters. In Proceedings of Game Developers Conference, pages 763–782, 1999.

[25] J. Shalf and E.W. Bethel. The grid and future visualization system architectures. IEEE Computer Graphics and Applications, 23(2):6–9, 2003.

[26] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. J. Nanosci. Nanotech. 8, pages 1551–1556, 2008.

[27] F. B. Vi´egas, M.Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Many Eyes: A site for visualization at internet scale. In Proceedings of IEEE InfoVis 2007, 2007.