# Salient Frame Detection for Molecular Dynamics Simulations

**Youngmin Kim[1], Robert Patro[2], Cheuk Yiu Ip[3], Dianne P. O'Leary[4], Andriy Anishkin[5], Sergei Sukharev[6], and Amitabh Varshney[7]**

**1,2,3,4,7 Department of Computer Science and
University of Maryland Institute for Advanced Computer Studies
College Park, MD 20742
E-mail:**
{ymkim,rob,ipcy,oleary,varshney}@cs.umd.edu

**5,6 Department of Biology, University of Maryland
College Park, MD 20742
E-mail:**
anishkin@icqmail.com
sukharev@umd.edu

## Abstract

Recent advances in sophisticated computational techniques have facilitated simulation of incredibly-detailed time-varying trajectories and in the process have generated vast quantities of simulation data. The current tools to analyze and comprehend large-scale time-varying data, however, lag far behind our ability to produce such simulation data. Saliency-based analysis can be applied to time-varying 3D datasets for the purpose of summarization, abstraction, and motion analysis. As the sizes of time-varying datasets continue to grow, it becomes more and more difficult to comprehend vast amounts of data and information in a short period of time. In this paper, we use eigenanalysis to generate orthogonal basis functions over sliding windows to characterize regions of unusual deviations and significant trends. Our results show that motion subspaces provide an effective technique for summarization of large molecular dynamics trajectories.

**1998 ACM Subject Classification** J. Computer Applications, J.2 Physical Sciences and Engineering

**Keywords and phrases** Saliency based analysis, Molecular Dynamics, Simulation

**Digital Object Identifier** 10.4230/DFU.Vol2.SciViz.2011.160

## 1 Introduction

Recent advances in acquisition and simulation techniques have resulted in immense time-varying datasets. These datasets are being used to study and explore a wide variety of phenomena in a diverse set of disciplines spanning life sciences to earth and space sciences. As the number and complexity of these datasets increases exponentially [9], it is becoming impractical to expect a domain expert to be able to look at such datasets manually. Automatic or semi-automatic tools to help humans discover scientifically interesting features are especially important for this reason.

Many illustration-based techniques have been proposed by several researchers [3, 10, 21] to summarize time-varying datasets such as ocean flow, volume, and human skeletons. The basic step for these illustration techniques is automatic detection of salient frames which

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 160–175
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

have interesting features. In the method of image saliency by Itti [8] or mesh saliency by Lee [13], they use a center-surround operator to identify the uniqueness of a pixel or a vertex with respect to its neighborhood. In this paper, we have decided to use a similar approach and define saliency as the uniqueness of a single frame with respect to its neighboring frames both forwards and backwards in time. Our collaborator, Dr. Sergei Sukharev's group at Biology Department at the University of Maryland, was interested in identifying the frames in molecular dynamics simulations, where the anomalies (kinks) in the secondary structures happen in the opening and closing simulations of thechannel [1]. We validate the effectiveness of our salient frame detection algorithm in this molecular dynamics simulation.

The rest of this paper is organized as follows. A review of related work is provided in Section 2. In Section 3, we formulate the relationship between one residue and the neighboring residues in space, and present an algorithm to detect saliency in time. Results are presented in Section 4. Performance considerations to improve the scalability of our approach for larger simulations are given in Section 5. Section 6 concludes this paper and discusses future work.

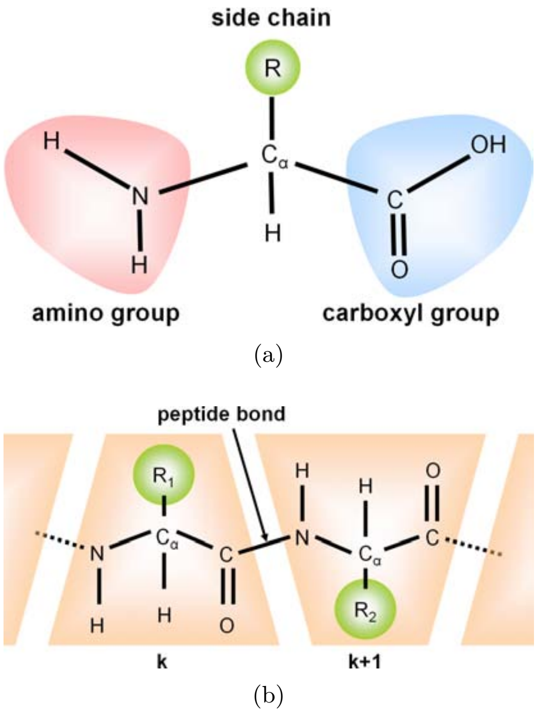
## 2 Background and Related Work

The goal of this paper is to detect salient frames in molecular dynamics simulations. This section briefly reviews some background in protein and ion channel structure and the related research in the area of motion analysis and visualization for time-varying 3D datasets.

### 2.1 Protein Structures

A protein structure is formed by a unique three-dimensional assembly of a specific polypeptide chain. Each polypeptide chain contains a particular sequence of serially linked amino acids. Figure 1(a) shows an amino acid which is composed of an amino group, a carboxyl group, and a side-chain, which are connected at the central $C_\alpha$ atom. When the carboxyl group of one amino acid reacts with the amino group of another amino acid, a peptide (i.e., amide) bond (Figure 1(b)) is formed by releasing a molecule of water ($H_2O$). This peptide bond is typically composed of four atoms (C, O, N, and H) which lie on a common plane due to the partial double bond characteristic at the CO-NH connection. Here, the recurring atomic array of N-$C_\alpha$-C(=O) from each amino acid of a polypeptide chain constitutes the protein *backbone*. By definition, the specific amino acid sequence for each polypeptide chain is the *primary* structure of the protein. Segments of polypeptides often fold locally into stable structures such as $\alpha$-helices or $\beta$-strands, each of which is called a *secondary* structure. An $\alpha$-helix is a right-handed coiled conformation, resembling a spring. $\beta$-strands connected laterally by three or more hydrogen bonds, form a generally twisted, pleated sheet.

The angle between two planes is referred as their *dihedral* angle. Figure 2(a) and (b) shows how we can compute the dihedral angle when there are four atoms which are not co-linear in 3D space. We first align the atoms $B$ and $C$ as shown in Figure 2(b). Then the dihedral angle corresponds to the angle measured in clockwise direction between the atom $A$ and the atom $D$. Similarly, for a sequence on a protein's polypeptide chain, backbone atoms (C, N, and $C_\alpha$) allow for three different dihedral angles of proteins as depicted in Figure 2(c): $\phi$ involving the backbone atoms C-N-$C_\alpha$-C, $\psi$ involving the backbone atoms N-$C_\alpha$-C-N, and $\omega$ involving the backbone atoms $C_\alpha$-C-N-$C_\alpha$. The planarity of the peptide bond usually restricts $\omega$ to be 180° or 0°. Thus the Ramachandran plot [19] considers two variable



(a)

(b)

**Figure 1** *Image (a) shows the structure of an amino acid. Image (b) shows a peptide bond formed by the reaction between a carboxyl group of one amino acid and an amino group of the other amino acid. Images are adapted from [6].*

dihedral (torsion) angles ($\phi$ and $\psi$) and shows possible combinations of these conformational angles of representative secondary structures in a polypeptide such as $\alpha$-helices or $\beta$-sheets.

## 2.2 Ion Channels

Ion channels are proteins that regulate the flow of ions into and out of the cells. Ion channels enable a very rapid flow of ions. In physiological conditions, MscS can provide for the flow of about a billion ions per second. Ion channel transitions are very fast – some opening for less than a millisecond before they close. This rapid and highly specific gating of ion channels is necessary for survival of cells. The ion channel kinetics impacts the speed at which ions flow across the cell membrane and the reaction time of a nerve or a muscle cell, and thus dictates the response time of the animal to the possible environmental dangers. An accurate understanding of the structural changes and functioning of ion channels is vital for therapeutic drug design. Nearly a third of the top 100 pharmaceutical drugs target ion-channels.

The bacterial mechanosensitive channel MscS and its eukaryotic homologs are principal turgor regulators in many walled cells. In bacteria, both free-living and pathogenic, these channels play critical roles of tension-driven osmolyte release valves thus allowing the organisms to avoid osmotic rupture in the event of abrupt medium dilution. MscS opening is driven directly by tension in the surrounding lipid bilayer and is accompanied by tilting of the pore-lining helices (TM3) which assume a kink-free conformation [2, 4]. When tension is released, the TM3 helices may buckle at two different hinge points, which defines the progression toward the closed state, as is shown in Figure 3(a). Thus, helical flexibility appears to

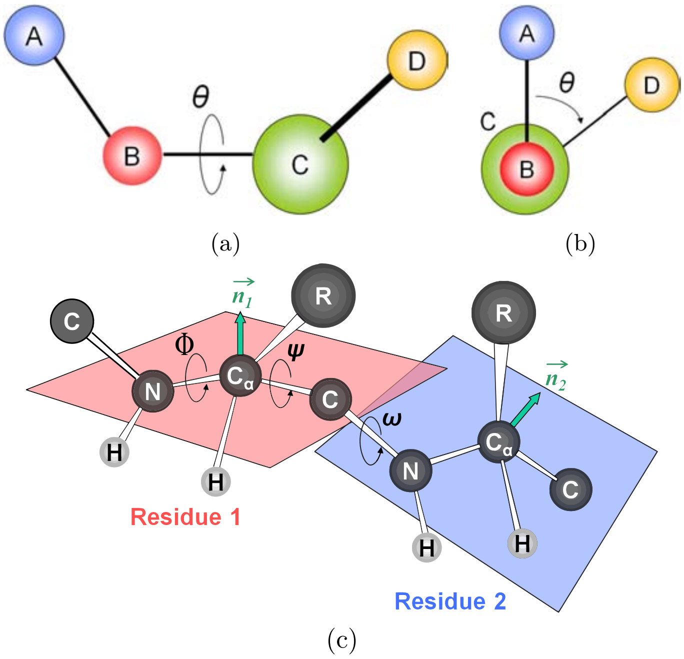


**Figure 2** *Images (a) and (b) show the computation of a dihedral angle between 4 atoms (A, B, C, and D). When we align the atom B and the atom C as shown in Image (b), the dihedral angle θ is defined as the angle between the atom A and the atom D in clockwise direction. Image (c) shows the dihedral angles ($\phi$ between C-N-$C_\alpha$-C, $\psi$ between N-$C_\alpha$-C-N, and $\omega$ between $C_\alpha$-C-N-$C_\alpha$) and the normal vectors ($\vec{n_1}$ and $\vec{n_2}$ on the planes defined using N-$C_\alpha$-C in residue 1 and residue 2, respectively. Images are adapted from [6].*

define the functional cycle of E. *coli* MscS. The major dataset analyzed consisted of two trajectories of atomic coordinates obtained from 4 *ns* steered simulations representing opening of wild-type and F68S mutant of E. *coli* MscS. The major goal was identification of frames in which conformations of helices deviated from the typical alpha-helical conformations.

## 2.3 Saliency-based Motion Analysis

Designers and artists have long used a single static image or a few images to illustrate dynamics of scenes for motion. They have depicted dynamics to facilitate visual communication in comic books and storyboards [14]. Recently, several graphics researchers [10, 17, 18] have proposed illustration-based techniques to depict the dynamics of time-varying data in a compact way. They use principles of visual art such as glyphs, and generate an image (or a few images) to summarize the time-varying data to facilitate visual communication. For instance, Joshi and Rheingans [10] have used illustration-based techniques such as speedlines, flow ribbons, and strobe silhouettes to convey change over time for a time-varying dataset. Nienhaus and Dollner [17] have used dynamic glyphs such as directed acyclic graphs and behavior graphs to provide further information about dynamics in the 3D scene.

A very interesting beginning in detecting salient frames for human skeleton datasets has been made by Assa *et al.* [3]. They generate an action synopsis for presenting the motion of a single skeleton-based character. They represent motion in affinity matrices, constructed from various aspects of a pose such as joint positions, joint velocities, joint angles, and joint angular velocities. They first define a vector $\mathbf{x}_\mathbf{a}^\mathbf{k}$ which represents an aspect $a$ of the pose at

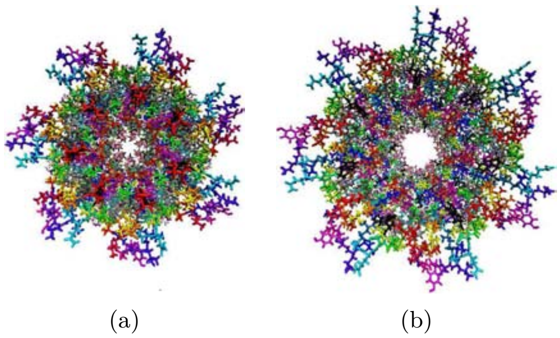


**Figure 3** *The images above show the closed (left) and open (right) conformations of the heptameric E.* coli *mechanosensitive channel MscS.*

frame $k$. Then, they compute the dissimilarity of the aspect $a$ between two given frames $i$ and $j$ by a simple distance measure to identify key poses. Finally, they compose these key poses into a single image by including the most significant poses.

There has been a significant increase in research activities related to the visualization of molecular dynamics simulations. Lampe *et al.* explore the use of a two-level hierarchical technique for the visualization of protein dynamics [12]. Recently, Krone *et al.* presented a method capable of visualizing molecular surface dynamics at interactive rates [11]. Tarini *et al.* [20] present a method to enhance shape perception in interactive molecular visualizations by employing ambient occlusion and edge cueing. Bidmon *et al.* present an informative and intuitive method for visualizing the motion of molecules around existing proteins using pathlines [5]. All of these papers [12, 11, 5] discuss methods for efficiently visualizing molecular dynamics, but do not detect key or salient frames in the simulation. Mehta *et al.* have explored approaches to the detection, classification and visualization of anomalous structures, such as defects in crystalline lattice structures [16, 15]. We are not aware of any research into the detection of salient frames for protein dynamics simulations.

## 3 Salient Frame Detection

In this paper, we characterize saliency as the uniqueness of a single frame with respect to its surrounding frames in time, and detect the salient frames for molecular dynamics simulations. A molecular dynamics simulation gives discrete samples of how a protein changes over a period of time. We are interested in identifying the time steps that highlight these changes by using subspace analysis. We analyze a particular time step $k$ in a molecular dynamics simulation with the following approach:

1. Select the residues of interest, this could be a subunit of a molecule or residues corresponding to an $\alpha$-helix of interest.
2. Model the angular relationship along the protein backbone for each time step $k$ as an affinity matrix $\mathbf{A}_k$.
3. Decompose the affinity matrix using SVD, $\mathbf{A}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T$
4. Analyze the salience, $s_k$, of the structure of the protein at time step $k$ with respect to the surrounding time steps.
5. Peaks of salience $s_k$ curve determine the best set of time steps.

Mechanosensitive ion channels play a critical role in transducing physical stresses at the cell membrane into an electrochemical response. The crystal structure of E. *coli* MscS has provided a starting point for detailed descriptions of its mechanism. Figure 3 shows the opening of the E. *coli* mechanosensitive ion channel that we will consider throughout this paper. There are 7 subunits in this ion channel, and all 7 subunits are topologically identical, but act relatively independently in the simulation. Out of 286 residues in each subunit, 175-residue N-terminal segments were included in simulations. To understand this mechanism, identifying the presence of kinks in $\alpha$-helices is critical because they have functional importance. Kink detection, however, is a tricky question because there are many factors involved. These include the state (ruptured or not) of the H bonds, local geometric information such as Ramachandran angles (torsion angles), and more global information such as the angles among multiple atoms.

In this section, we formulate the relationship between one residue and the neighboring residues spatially, and present an algorithm to detect saliency in time. Our framework encompasses the global and local geometric properties of backbone residues in a molecular dynamics simulation.

## 3.1 Construction of Affinity Matrices in Space

We explore the relationship between one residue and the neighboring residues to detect the changes in $\alpha$-helices. The straightening and buckling of $\alpha$-helices are interesting because they appear in many simulations of ion channels and are believed to be correlated with conformational states of the whole channel. There are many ways to define the relationship among residues, but we believe the angles in backbone atoms would be one of the best ways since backbone atoms are much more stable in their positions than side-chains. As a Ramachandran plot suggests, we could have measured torsion (dihedral) angles and conjectured the changes of secondary structures for each residue. However, analysis of Ramachandran angles only considers very local properties inside a residue, and does not encompass the global geometric property among a sequence of residues. Instead, we use the relative angles between one $C_\alpha$ ($\alpha$-carbon) and other $\alpha$-carbons within a cutoff distance, $r_s$. The cutoff *distance* refers to the difference in index between the $C_\alpha$ currently being considered and its neighboring residues along the chain. A good choice is $r_s = 5$, because on average, $\alpha$-helices turn once every 3.6 amino acids. Considering ±5 amino acids forwards and backwards should cover, in total, about 3 turns in $\alpha$-helices, which is a sufficient scale for kink detection. Alternatively, instead of imposing a hard cutoff, we can use a Gaussian cutoff.

Molecular dynamics simulations give us a trajectory file which holds all the atom positions in 3D space for every frame $k$. Since three non-co-linear points in 3D space define a plane, the positions of N-$C_\alpha$-C atoms in each residue can define a plane and its normal vector $\vec{n}$ as shown in Figure 2(c). We compute normal vectors ($\vec{n_i}$) to the planes formed by these N-$C_\alpha$-C atoms in residues ($R_i$) for every frame $k$.

Specifically, we model the interactions amongst neighboring amino acids for time step $k$ by an affinity matrix $\mathbf{A}_k$. Each entry $a_{ij}$, of the matrix $\mathbf{A}_k$ models the strength of interaction between amino acid residues $i$ and $j$. As discussed above, each amino acid residue, $i$, is associated with the vector $\vec{n_i}$.

$$a_{ij} = \begin{cases} \frac{\langle \vec{n_i}, \vec{n_j} \rangle}{\frac{r_s}{2}\sqrt{2\pi}} e^{-\frac{(j-i)^2}{r_s^2}} & \text{if } |j-i| < r_s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. This allows us to generate the

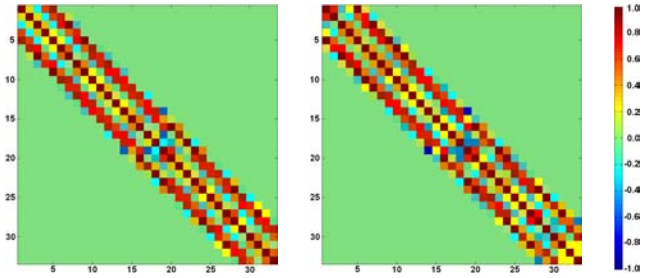


**Figure 4** *Visualization of affinity matrices computed from the first and the second frames for* 33 *residues of the subunit* 1 *for E.* coli *MscS (shown in Figure 3) when the cutoff distance, $r_s = 5$ is used.*

affinity matrix $\mathbf{A}_k$ as:

$$A_k = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm} \end{bmatrix} \tag{2}$$

Figure 4 visualizes the affinity matrices from the first and the second frames for 33 residues (from residue 94 to residue 126) of the subunit 1 for the molecule shown in Figure 3.

## 3.2 Saliency Detection among Neighboring Affinity Matrices

Our affinity matrix in equation 2 represents the geometric relationship among neighboring residues. This angular relationship cannot be represented by a single vector $\mathbf{x}$ as in [3]. Therefore, the dissimilarity between two given frames $i$ and $j$ should be computed as the difference between two affinity matrices $A_i$ and $A_j$.

A naïve approach to comparing two affinity matrices $A_i$ and $A_j$ is to directly measure the difference. For example, one might compute $\|A_i - A_j\|_F$, where $\|\cdot\|_F$ is the Frobenius norm, and for a matrix $M$ is defined as:

$$\| M \|_F = \sqrt{\sum_{a=1}^{m} \sum_{b=1}^{n} |m_{ab}|^2}$$

where $m_{ab}$ is the entry in the $a^{\text{th}}$ row and $b^{\text{th}}$ column of $M$.

However, as the affinity matrices are constructed directly from the simulation data, they encode high frequency changes in atom positions that are the result of Brownian motion. Yet, we wish to identify large-scale conformational changes in the molecule. For our purposes, the high frequency information in the affinity matrix which results from Brownian motion is essentially noise. Thus, when comparing two affinity matrices, we wish to ignore such noise and to consider only more significant changes. For this purpose, we employ the Singular value decomposition (SVD), the computation of which can be found in the standard text by Golub and Van Loan [7]. The SVD factorizes a given $m \times n$ matrix $A$ into three matrices: $A = U\Sigma V^T$, where $U$ is an $m \times m$ orthogonal matrix ($UU^T = I$ and $U^TU = I$), $\Sigma$ is $m \times n$

diagonal matrix with non-negative numbers, and $V^T$ is the transpose of an $n \times n$ unitary matrix $V$. There are three properties of the SVD of which we shall take advantage:

1. $U$ and $V$ are a set of orthonormal basis vectors (singular vectors).
2. The diagonal entries in $\Sigma$ (called singular values), are sorted in non-increasing order and indicate the importance of the corresponding basis vectors.
3. $\hat{A} = U\hat{\Sigma}V^T$, where $\hat{\Sigma}$ is the same as $\Sigma$ except that all but the $d$ largest singular values have been set to zero, is the best rank-$d$ approximation to $A$ in the sense that:

$$\hat{A} = \min_{\{M|\text{rank}(M)=d\}} \|A - M\|_F$$

In $\alpha$-helices, backbone atoms (C, O, N, and H) are much more stable than side-chains because of the H bonds. However, there are still significant random vibrations in the positions of backbone atoms over time due to the effects of Brownian motion. By using SVD to obtain a lower-rank approximation of our original affinity matrices, we hope the to reduce or eliminate the effects of such vibrations by ignoring the contributions of the highest frequency singular vectors, where we expect their contributions to reside. This is why we use SVD as opposed to other methods for computing the uniqueness of the affinity matrix $A_i$ relative to the affinity matrix $A_j$.

### Uniqueness of the affinity matrix $A_i$ relative to the affinity matrix $A_j$:

We use the following procedure to compare the affinity matrices $A_i$ and $A_j$ for frames $i$ and $j$. We perform a singular value decomposition of $A_i$ to obtain $A_i = U_i \times \Sigma_i \times V_i^T$. This returns the basis vectors as the column vectors in $U_i$. Since the basis vectors are sorted by their importance in SVD decomposition, we can obtain a reduced (rank $d$) matrix $\hat{U}_i$ by taking the first $d$ basis vectors in $U_i$. For the $j$-th frame $A_j$, we use these $d$ basis vectors to best approximate it. For this, we project $A_j$ to the low-dimensional subspace spanned by the $d$ basis vectors as: $W_{i,j} = \hat{U}_i^T \times A_j$. This gives us the weight matrix $W_{i,j}$ for the $d$ basis vectors. We use this weight matrix to approximate $A_j$ by: $\hat{A}_j = \hat{U}_i \times W_{i,j}$. Finally we compute the root mean square error ($\epsilon_{ij}$) between $\hat{A}_j$ and $A_j$: $\| \hat{A}_j - A_j \|_F$.

To determine $d$, we use a user-specified parameter $\tau$ and choose $d$ to be the largest integer such that $\delta_i = \frac{\|\hat{A}_i^d - A_i\|_F}{\|A_i\|_F} < \tau$; where $\hat{A}_i^d$ is the rank $d$ approximation of $A_i$ obtained using the truncated singular value decomposition. Note that $\delta_i^2$ can be easily computed as $\sum_{j>d} \sigma_j^2 / \sum_j \sigma_j^2$, where $\sigma_j$ is the $j$th largest singular value.

### Saliency Value $s_i$ for the frame $i$:

To compute the uniqueness of a frame $i$ relative to other frames $j$, we avoid considering all possible pairs $(i, j)$. Instead, we consider neighboring frames $j$ where $|i-j| \leq r_t$. Throughout this paper, we use $r_t = F/10$, where $F$ is the total number of frames. The final saliency value $s_i$ is the average of the errors $\epsilon_{ij}$ in neighboring frames of $i$:

$$s_i = \frac{\displaystyle\sum_{|j-i| \leq r_t} \epsilon_{ij}}{F_i} \tag{3}$$

where $F_i$ is the number of frames whose distance from the frame $i$ is less than or equal to $r_t$. Figure 5 shows the graph for these saliency values in blue.

# 4 Results

We have compared our detected salient frames with the ones identified independently by our collaborators (biology scientists) for molecular dynamics simulations. Figure 5 shows the five most salient frames detected by our method for the subunit 4 in the E. *coli* mechanosensitive ion channel in Figure 3. The frames 5, 26, 30, and 34 which have been detected by our method are the same or very close to the frames 3, 24, 26, 30, and 35 with changes in the kinks, which were detected manually by our collaborators. The frame 39 detected by our algorithm is not close to any frame detected manually by our collaborators, but it had the lowest saliency value among the five most salient frames. Generally, kinks change towards the end of this simulation, and our method successfully detects these important frames.

Figure 6 shows the five most salient frames detected by our method for the subunit 1 in the ion channel shown in Figure 3. This subunit is topologically identical to the subunit 4, but acts differently in the simulation. Therefore, it results in different salient frames (frames 11, 19, 21, 35, and 39) as shown in Figure 6. Our collaborators identified frames 2, 18, 20, 23, 35, 36, and 39 as being salient. Figure 7 shows the six most salient frames detected by our method for the subunit 4 in the symmetry annealing of MscS F68S mutant. In this molecular dynamics simulation, residue 68 was mutated to another, serine, which has very specific consequences for channel inactivation in real experiments. As changes in the kinks occur more frequently than the previous simulations, we observe a larger number of salient frames than in the previous cases. Our collaborators have manually identified frames 2, 4, 18, 34, and 38 as being salient. Among these, frames 2, 4, 18, and 38 are the same or close to the frames 1, 5, 18, and 39 detected by our algorithm, and the remaining frame 34 also exhibits a relatively high saliency value as shown in Figure 7.

# 5 Performance Considerations

Performance considerations need to be taken into account in order to make our approach feasible for large datasets. The analysis of a particular simulation may require us to consider thousands of amino acid residues, and thousands of frames. The running time of our algorithm is dominated by the computation of the SVD for each affinity matrix. If we consider $r$ residues and $F$ frames, this leads to a worst case complexity in $\mathrm{O}(r^3 F)$. However, since our affinity matrices are highly sparse, and since we usually require use a rank $d$ approximation of these matricies with $d \ll r$, our algorithm is far more efficient in practice. Below, we detail a number of optimizations we implemented to allow our approach to scale to problems of the size we need to consider.

## 5.1 Maintaining Only Local Data

For very small data sets, it is possible to first calculate the requisite information for every time step of the simulation and then to analyze the errors considering the window centered around each. Though this approach eases the implementation of our algorithm somewhat, it scales poorly to even moderately sized data. We typically expect molecular dynamics simulations to run for many time steps. Yet, we will not be able to store the necessary information for all time steps in random access memory.

To overcome this difficulty, the implementation of our algorithm stores only local data that is relevant to the analysis of a time step, $k$, about which the current window is centered. In particular, for time step $k$, it is necessary to compute $\mathbf{A}_k$, $\hat{\mathbf{U}}_k$ and for all other time steps $\ell$ in the window, it is necessary to compute $\mathbf{A}_\ell$. This information is sufficient to compute a saliency measure for time step $k$. Storing only the information necessary to analyze the current time step implies that the memory requirements can be made independent of the length of the simulation. Ideally, the size of a window is based on the physical timescale over which actions of interest are expect to occur in the molecule, and it varies independently of the number of time steps in the simulation.

## 5.2 Data Reuse

A second practical consideration we make in our implementation of our algorithm is the reuse of data to avoid redundant computation. If the algorithm is implemented in a naïve fashion (relevant analysis data for the entire simulation is computed at once) then this is trivially achieved. However, even when we consider only the local data, in the window $W_k$, relevant to the analysis of a time step $k$, it is possible to reuse many of the computed quantities when considering the window $W_{k+1}$, centered around the next time step, $k+1$. When the sliding window is moved forward by a single time step, from $W_k$ to $W_{k+1}$, all but the leftmost of the affinity matrices from $W_k$ remain relevant to the analysis of the new window. Further, since all but one of the affinity matrices from the previous window are reused, only a single new affinity matrix, corresponding to the rightmost time step in $W_{k+1}$ need be computed. Finally, the basis vectors, $\hat{\mathbf{U}}_k$, may be discarded while $\hat{\mathbf{U}}_{k+1}$ will be computed. Thus, by reusing relevant data as the sliding window proceeds forward along the time steps of a simulation, we can ensure that, despite the fact that we only store window-local data, each affinity matrix and approximate basis is computed only once.

## 5.3 Exploiting Sparsity

Even if we only consider storing window-local data, memory requirements might still be exorbitant if we need to consider many amino acid residues for each time step. This is due to the fact that we will require the storage of an affinity matrix for each time step in the current window. However, since our affinity matrix considers only local interactions (residues within 5 units of each other along the protein backbone), the matrix itself is very sparse. In the affinity matrix of a given time step, $k$, each row will have, at most, 11 non-zero entries. Thus, by using a sparse matrix structure the memory requirements for storing an affinity matrix can be made linear, rather than quadratic, in the number of considered amino acid residues. This enables us to consider many residues for each time step while keeping feasible memory requirements.

## 5.4 Iterative SVD Transform Using Spectral Shift

The singular value decomposition is the most computationally intensive step of our algorithm. Yet, even this step of the algorithm can be optimized significantly by obtaining singular vectors iteratively. When obtaining an approximate basis, $\hat{\mathbf{U}}_k$, for the affinity matrix $\mathbf{A}_k$, we need enough singular vectors so that we can represent $\mathbf{A}_k$ with sufficient accuracy. However, the number of basis vectors required for the desired accuracy is often significantly less than the number of columns ( or rows ) of $\mathbf{A}_k$. Thus, it is wasteful and unnecessary to perform a full SVD of $\mathbf{A}_k$. Most SVD implementations allow the user to request only the $D$ most significant singular vectors. Unfortunately, we do not know, a priori, the number of vectors that will be required to reach our desired error threshold. Additionally, most sparse SVD implementations exhibit another behavior that is undesirable. Namely, the running time of the algorithm is super-linear in the number of requested singular values/vectors. We adopt the approach suggested by Vallet and Levy [22] to overcome both of these difficulties simultaneously.

To overcome the aforementioned difficulties, we make use of the ability (available in most SVD implementations) to request the singular values and corresponding singular vectors that are closest to a particular spectral shift value, $\sigma$. We request a fixed number, $d = 50$, of singular values for each call to the SVD procedure. When we obtain the results, we find the largest ($\sigma_u$) and smallest ($\sigma_l$) singular values. Then, we compute a spectral shift $\sigma_s = \sigma_l + \lambda(\sigma_u - \sigma_l)$ for the next invocation of the SVD procedure. Here, $\lambda$ is a small scalar value (we use $\lambda = 0.2$), and $\sigma_s$ is computed so that there is overlap between the spectra returned by consecutive calls to SVD. Since we only request $d$ singular vectors per invocation of the SVD procedure, we avoid the super-linear runtime in number of requested singular vectors. Furthermore, after each iteration, we obtain a more complete set of basis vectors for the affinity matrix; driving down the residual error. Thus, we can compute the residual error after each iteration, subsequently ensuring that the total number of singular vectors we obtain from the SVD is never more than $d-1$ in excess of the amount required to satisfy our error threshold.

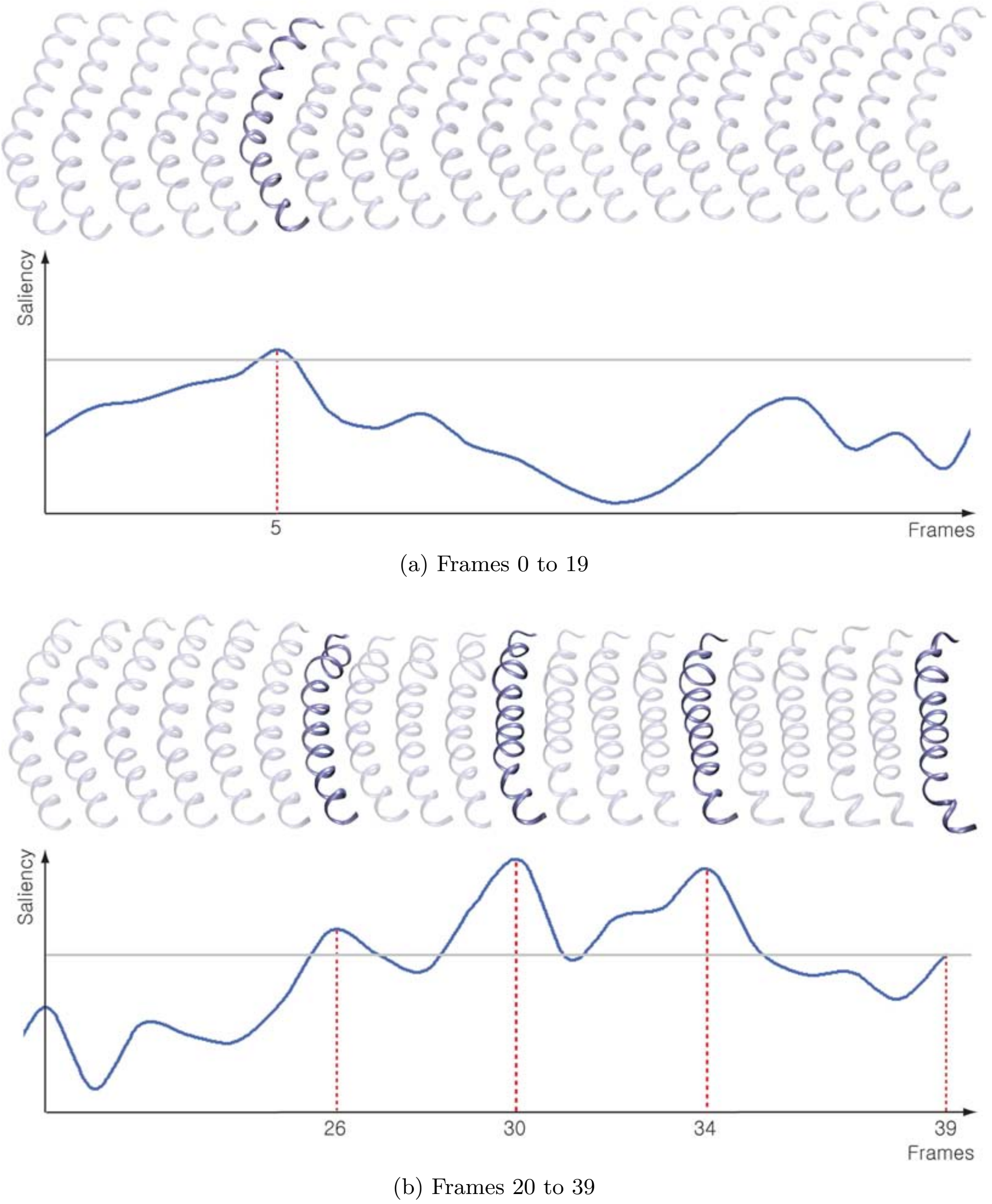
## 6 Conclusions and Future Work

In this paper, we have detected salient frames for molecular dynamics simulations. We have introduced the notion of saliency in time, and successfully identified most of the key frames which have changes in the kinks (i.e. appearance or disappearance of a kink) for E. *coli* channel. We believe that our method can enable researchers to focus on the important frames for further analysis of the dataset.

We currently consider the angles between the normal vectors defined by the planes of different residues in $\alpha$-helices, and identify the anomalies (kinks) in the secondary structures for an E. *coli* channel. However, it makes sense to explore other structural properties of the molecule as well. For example, we could consider quantities like the rotational angles between residues or the derivatives of such quantities over time. It will be interesting to see how affinity matrices based on other quantities compare to the ones we have chosen for this work. Also, it will be interesting to explore how the approach detailed in this work might be generalized to other types of time-varying data. We believe this framework can be easily extended to encompass salient features in other time-varying simulations by changing the way we construct affinity matrices to address other needs by scientists or domain experts.

## Acknowledgment

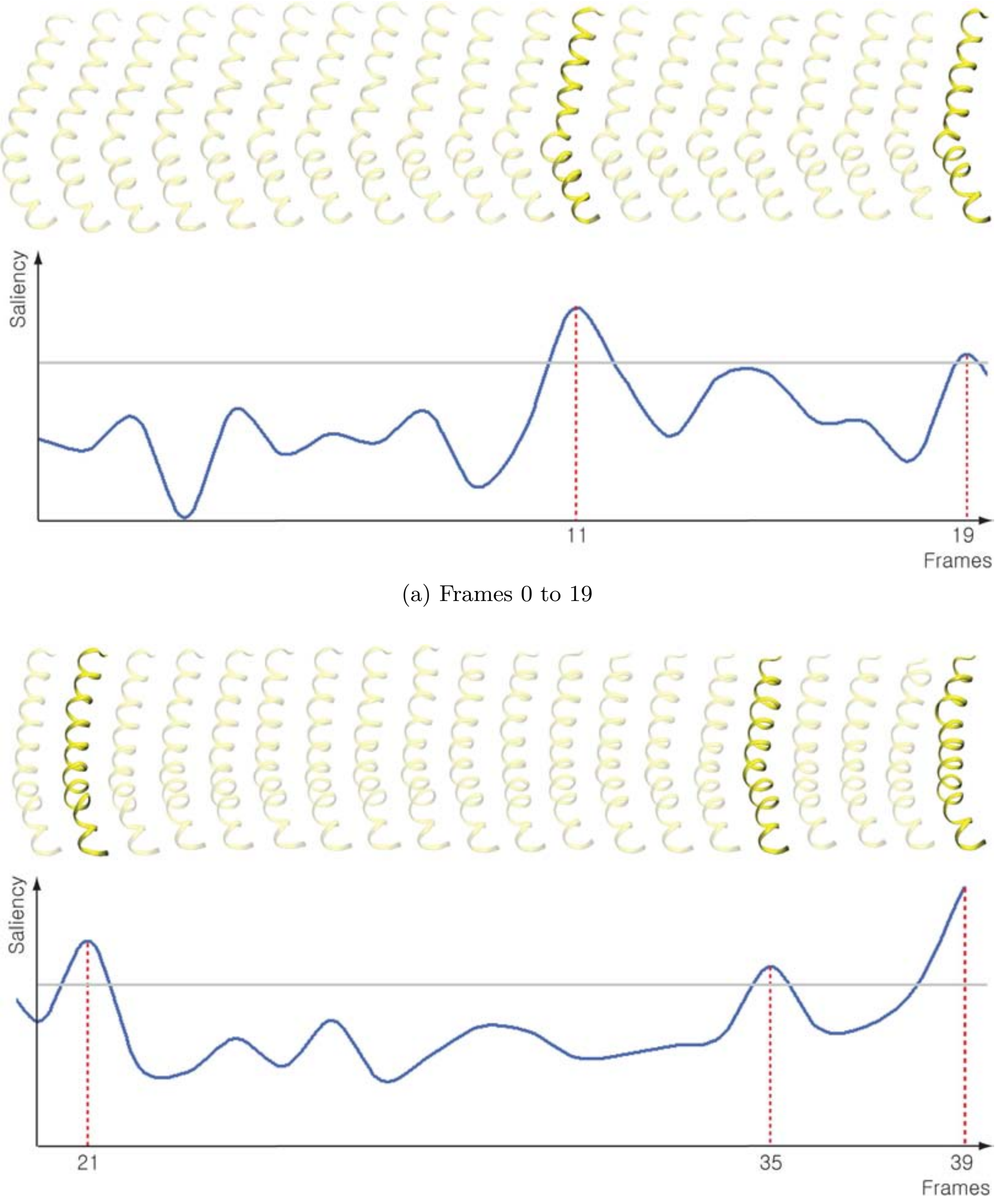
We gratefully acknowledge David Jacobs, David Mount, and Yoonkyung Kim for several valuable discussions. This work was supported in part by the NSF grants: CCF 05-41120, CCF04-29753, CNS 04-03313, and IIS 04-14699 and the NIH grants R01GM075225 and 2R01 NS03931405A. We would also like to thank the anonymous reviewers for their detailed and highly constructive comments that were extremely helpful.



(a) Frames 0 to 19

(b) Frames 20 to 39

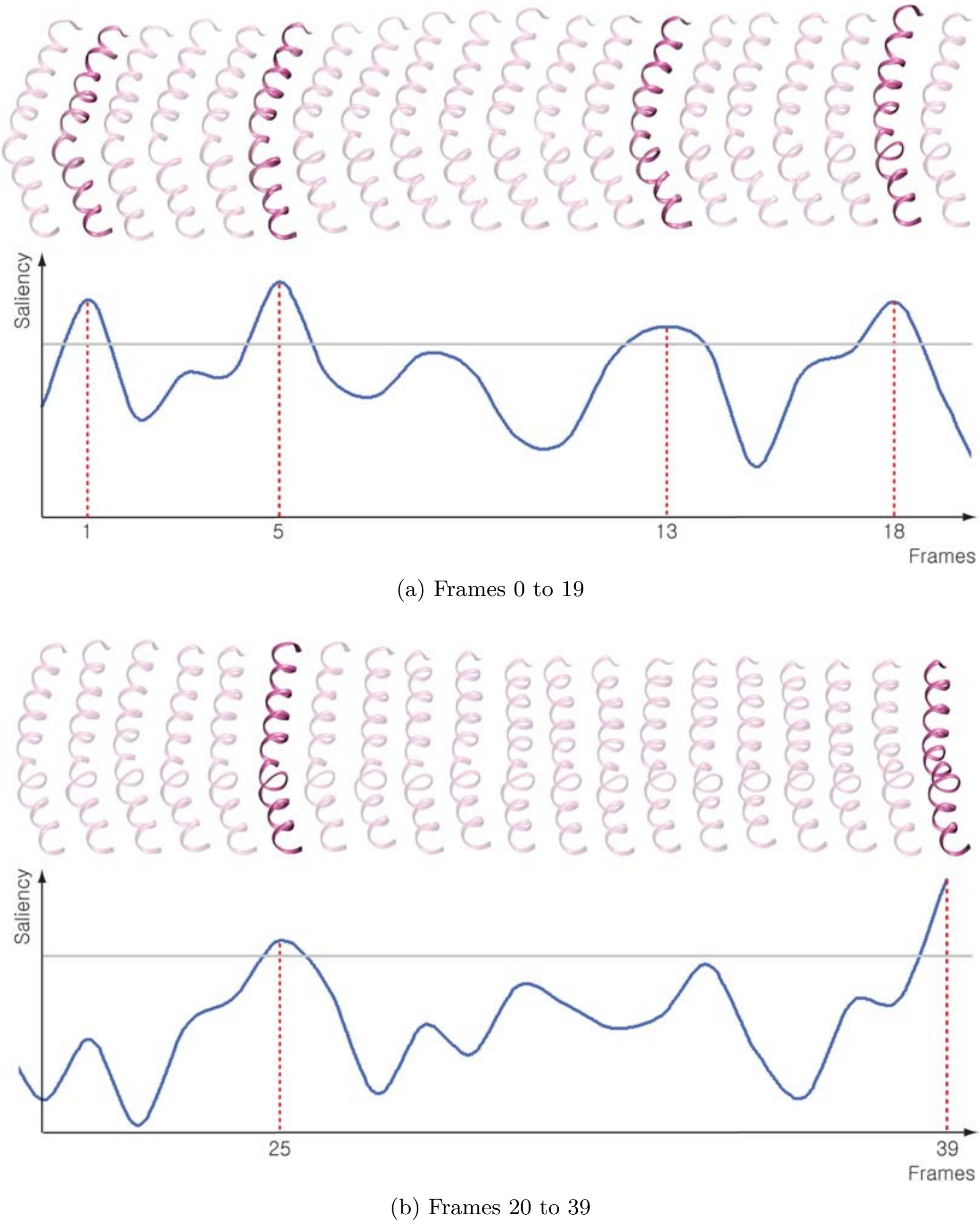
**Figure 5** *Five most salient frames detected by our method for the subunit 4 in the E.* coli *ion channel (MscS) in Figure 3. The changes in the kinks are detected towards the end of this simulation, and our method successfully detects some of the most important frames.*



(a) Frames 0 to 19

(b) Frames 20 to 39

**Figure 6** *Five most salient frames detected by our method for the subunit* 1 *in the E.* coli *ion channel (MscS) in Figure 3. This subunit is topologically identical to the subunit* 1*, but acts differently in the simulation.*



(a) Frames 0 to 19

(b) Frames 20 to 39

**Figure 7** *Six most salient frames detected by our method for the subunit* 4 *in the other molecular dynamics simulation, showing the symmetry annealing of MscS F68S mutant – the residue* 68 *was mutated to another, serine, which has very specific consequences for channel inactivation in real experiments.*

## References

1 B. Akitake, A. Anishkin, N. Liu, and S. Sukharev. Straightening and sequential buckling of the pore-lining helices define the gating cycle of mscs. *Nature Structural and Molecular Biology*, 14(12):1141–1149, 2007.

2 Andriy Anishkin and Sergei Sukharev. State-stabilizing interactions in bacterial mechanosensitive channel gating and adaptation. *The Journal of Biological Chemistry*, 284(29):19153–19157, Jul 2009.

3 J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: pose selection and illustration. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 24(3):667–676, 2005.

4 V. Belyy, A. Anishkin, K. Kamaraju, N. Liu, and S. Sukharev. The tension-transmitting 'clutch' in the mechanosensitive channel mscs. *Nat Struct Mol Biol*, 2010.

5 Katrin Bidmon, Sebastian Grottel, Fabian Bös, Jürgen Pleiss, and Thomas Ertl. Visual abstractions of solvent pathlines near protein cavities. *Computer Graphics Forum (EuroVis 2008 Special Issue)*, 27(3):935–942, 2008.

6 C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing, Inc., second edition, 1999.

7 Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

8 L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 20(11):1254–1259, 1998.

9 C. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T. S. Yoo. NIH-NSF visualization research challenges report. Technical report, Mitsubishi Electric Research Laboratories, 2006. Computing in Science and Engineering,.

10 A. Joshi and P. Rheingans. Illustration-inspired techniques for visualizing time-varying data. In *IEEE Visualization*, pages 679–686, 2005.

11 Michael Krone, Katrin Bidmon, and Thomas Ertl. Interactive visualization of molecular surface dynamics. *IEEE Trans. on Visualization and Computer Graphics*, 15(6), 2009.

12 Ove Daae Lampe, Ivan Viola, Nathalie Reuter, and Helwig Hauser. Two-level approach to efficient visualization of protein dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1616–1623, 2007.

13 C. H. Lee, A. Varshney, and D. Jacobs. Mesh saliency. *ACM Trans. on Graphics (Procs. ACM SIGGRAPH)*, 24, No. 3:659 – 666, 2005.

14 S. McCloud. *Understanding Comics – The Invisible Art*. Harper Perennial, 1994.

15 Sameep Mehta, Steve Barr, Alex Choy, Hui Yang, Srinivasan Parthasarathy, Raghu Machiraju, and John Wilkins. Dynamic classification of defect structures in molecular dynamics simulation data. In *Proceedings of SIAM on Data Mining*, 2005.

16 Sameep Mehta, Kaden Hazzard, Raghu Machiraju, Srinivasan Parthasarathy, and John Wilkins. Detection and visualization of anomalous structures in molecular dynamics simulation data. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 465–472, Washington, DC, USA, 2004. IEEE Computer Society.

17 M. Nienhaus and J. Dollner. Depicting dynamics using principles of visual art and narrations. *IEEE Computer Graphics and Applications*, 25(3):40–51, 2005.

18 G. Pingali, A. Opalach, Y. Jean, and I. Carlbom. Visualization of sports using motion trajectories: Providing insights into performance, style, and strategy. In *Proceedings Visualization 2001*, pages 75–82, 2001.

19 G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Mol. Biol.*, 7:95–99, 1963.

20 Marco Tarini, Paolo Cignoni, and Claudio Montani. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Transaction on Visualization and Computer Graphics*, 12(6), sep/oct 2006.

21 G. Turk and D. Banks. Image-guided streamline placement. In *Proceedings of SIGGRAPH 1996*, pages 453–459, 1996.

22 Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proceedings Eurographics)*, 2008.