Compression in Molecular Simulation Datasets

Anand Kumar[1], Xingquan Zhu[2], Yi-Cheng Tu[1], and Sagar Pandit[3]

[1] Department of Computer Science and Engineering, University of South Florida, Tampa, FL - 33620, U.S.A.
{akumar8,ytu}@cse.usf.edu
[2] Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL - 33431, U.S.A.
xzhu3@fau.edu
[3] Department of Physics, University of South Florida, Tampa, FL - 33620, U.S.A.
pandit@cas.usf.edu

Abstract. In this paper, we present a compression framework, for molecular dynamics (MD) simulation data, which yields significant performance by combining the strength of principal component analysis (PCA) and discrete cosine transform (DCT). Though it is a lossy compression technique, the effect on analytics performed on decompressed data is very minimal. Compression ratio up to 13 is achieved with acceptable errors in results of analytical functions.

Keywords: molecular simulations, encoding, data compression, compression ratio, principal component analysis, discrete cosine transform.

1 Introduction

Many scientific disciplines, such as biochemistry, astronomy, and material sciences, are undergoing a radical change in research methodology from conducting "wet-bench" experiments to performing computer simulations. As a result, the particle simulations (PS) have seen tremendous efficiency improvements in the last decade. In PS, the system of interest (e.g., a protein and its environment) is studied as a collection of large number of basic components (e.g., atoms) whose behavior can be completely described by classical physics. Often such simulations generate spatio–temporal data that are in tera to peta bytes in size [1]. With increasing size of data, the problem of efficient storage and transfer persists. This paper addresses these issues in spatio-temporal data generated from PS applications by proposing effective data compression techniques.

1.1 Motivation and Contributions

The PS data is obtained from simulation results of a biological, physical or chemical phenomenon. In these systems, all individual atoms (we use atom and particle interchangeably) together represent large biological structures. Thus,

C. Sun et al. (Eds.): IScIDE 2013, LNCS 8261, pp. 22–29, 2013.

providing nano-scopic description of biological process. The simulation consists of measuring properties such as, 3D location of the atoms, velocity, charge, mass etc., at very small intervals of time (pico-seconds). Measurements of all atoms taken at a time instant, called snapshot (or *frame*), are stored on to computer disk. Considering the simulation for few microseconds, the data generated can easily reach terabytes. Since the simulation is generally done in large computer clusters, data needs to be transferred to a storage server. On the server end, data analytics face bottleneck due to limited I/O bandwidth. The above facts necessitate the compression of data for better utilization of the storage devices and network transfer bandwidth. The MD and the data management communities have been primarily focused on the high-performance computing, visualization and simple data management, thus left the problem of MD data compression inadequately addressed.

Traditional dictionary based approaches have inherent disadvantages in compressing particle simulation data: (1) compressed data size can still be large; and (2) whole data is scanned before starting compression. In addition, decompression becomes impossible if one or few data bytes are corrupted or damaged.

Contributions: The existing compression methods do not consider the temporal locality of the atoms for compression, which leaves a significant amount of redundancy in the compressed data. Our technique is designed to address these problems and meet the following four goals.

- High compression ratio (> 5), without noticeable errors is desirable, and dynamic error control to meet predefined requirements of compression quality.
- Error tolerance in compressed data and largely de-compressible. Even if some parts are corrupted, the errors are not propagated across many data frames.
- Access to random frames is allowed, without decompressing the whole data. We achieve random access to a small group of frames.
- Balanced compression across different dimensions of the data.

In our framework, the MD data are first transformed, using PCA (Section 2.1), from the generic 3D coordinate space to another 3D eigen space, with the dimensions sorted in decreasing importance levels in capturing the variance of the atoms' movements. In the eigen space, the DCT is applied (Section 2.2) to achieve lossy compression across a window of consecutive frames. The lossy compression does not affect the results (Section 3) of the analytics that are often executed on molecular simulation data [2,3,4]. The combination of the PCA and DCT ensures that our framework can achieve aforementioned goals.

1.2 Related Work

Popular compression tools like WinZip and Gzip use dictionary based methods (such as Ziv-Lempel [5]). Statistical methods like Huffman encoding and arithmetic encoding [5] are used in compression of the multimedia data. These methods are either suitable for large text data or data that exhibit certain statistical properties. The resulting size of the compressed data can still be large.

Thus adding high cost to I/O and network transfers. There have been efforts to process approximate analytics using histograms [3], wavelets [4], and random sampling [6]. In these methods, the focus was mainly on the efficient data analysis while minimizing the I/O during execution time. A detailed survey of the traditional data compression techniques is provided by Salomon *et al.* [5].

A combination of different encoding-based compression techniques for molecular dynamics (MD) data is presented by Omletchenko *et al.* [2]. The technique utilizes spatial locality of the atoms using oct-tree index and space-filling curve (SFC), before encoding the data. Another approach to compression of trajectories and data management is provided by Essential Dynamics (ED) tool [7]. It is similar to compression of data using principal component analysis (PCA) [8], in which the trajectory pattern and the number of eigenvectors determine the error produced in the uncompressed data. These tools do not consider temporal locality of the atoms for compression. Hence, the achievable compression is limited. In molecular dynamics simulations, most atoms move along a trajectory of their own that changes very little over time. There is a huge scope for achieving high compression if temporal locality is considered. We attempt to achieve better compression in our approach by considering these features.

2 Compression Framework

In this section, we discuss the the basic framework of compression along with the encoding techniques used. Fig. 1 shows the framework of the proposed compression method. Our main theme is to employ PCA to transform data to another space, on which lossy compression can be achieved using DCT. Due to space limitations we omit some details of the modules.

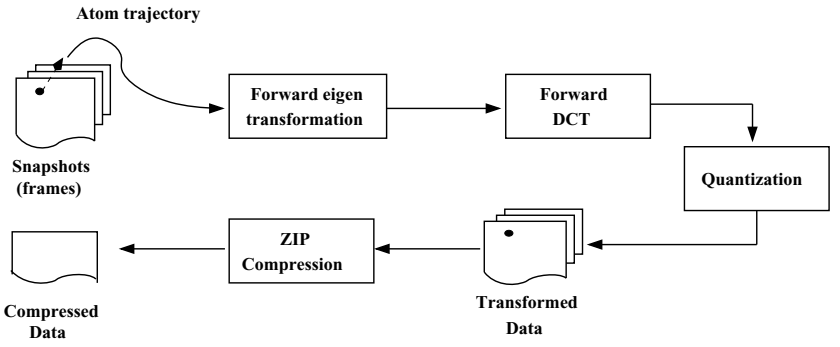


Fig. 1. The framework for compression of particle simulation data

2.1 Transforming Data to Eigen Space

The measurement values captured during molecular simulation change over time as the data is stored after each snapshot. The measurements depict changes in

certain properties of the atoms in the simulation. As the MD data is of large volume, we compress a group of snapshots (frames) called window. The number of frames in a window can be controlled depending on the availability of computational resources and the level of errors we are willing to tolerate in the final data. We apply orthogonal linear transformation to the data using principal component analysis (PCA) [8]. For each atom (as in Fig. 2) we collect its locations across a number of adjacent frames, $f_1, f_2, \ldots, f_n$, each of which contributes a 3D location (l_x, l_y, l_z). PCA transforms data to a new coordinate system (l_p, l_q, l_r) such that the greatest variance is observed on the first coordinate l_p after projection of the data. The coordinates are ordered in decreasing order of the variance of the projection, which is obtained by ordering the eigen values and corresponding eigenvectors of the data. The transformation $\mathbf{F}$ of given data $\mathbf{D}$, with eigenvectors $\mathbf{E}$ in decreasing order of eigen values, using PCA is given by equation (1).

$$\mathbf{F}^T = \mathbf{D}^T\mathbf{E} \tag{1}$$

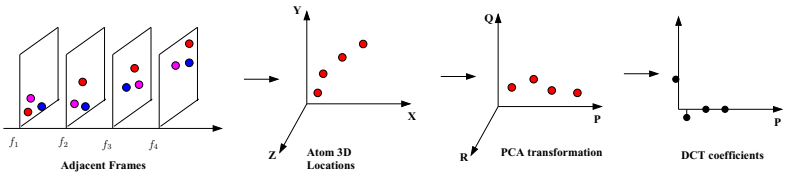


Fig. 2. Intuition behind compression: Performing principal component analysis (PCA) followed by discrete cosine transform (DCT) on data

The eigen values and eigenvectors can be found from the covariance matrix of the data in the window. The original data $\mathbf{D}$ can be obtained back from the transformation using equation (2).

$$\mathbf{D}^T = \mathbf{E}^{-1}\mathbf{F} \tag{2}$$

The eigen space transformation aligns the data along eigenvectors based on the variance observed in the data. By transforming the data to the new coordinate system (P, Q, R), we can pick few dimensions that represent the complete data (compressed) while maintaining low errors (in decompressed data). This is lossy compression. In our approach we transform the data into new coordinates to obtain the different directions of variance. These directions are then compressed independently using DCT. Each dimension uses different DCT parameters based on the degree of variance for compression. This gives data with minimal loss to achieve better performance of the DCT compression (Section 2.2).

2.2 Computing DCT of Transformed Data

The eigen analysis step transforms data into new dimensions [8]. We encode every dimension (attribute/measure) of the atoms separately. However, the eigen

space transformation is applied to 3D trajectories of each atom separately. Finally, we apply the DCT on each dimension P, Q, R (after dropping unnecessary dimensions) of the transformed data. By transforming the data to the new coordinate system, we are trying to reduce the errors in all dimensions that may occur in performing the DCT step. This makes recovering DCT coefficients more accurate during decompression, as the coefficients in the beginning contain more accurate information.

The atom localization property is utilized by applying the DCT to a constant number of adjacent frames, say N. Given a window of size N (frames), we compress single measurement of every atom individually. The trajectory of length N is transformed using DCT. Such transformation is applied to trajectories of all atoms in the window. Now, the DCT data $\mathbf{C}$ is quantized to reduce the number of bits required to store the coefficients. The coefficients can be eliminated by assigning very low weights to the high frequency components. After all the data is processed, we apply ZIP compression on the DCT data. As a lossless compression method, the ZIP compression removes the redundant information present in the data. This gives the final compressed data. The combination of the PCA and DCT in the framework ensures that (1) compression is balanced across all dimensions, (2) error can be controlled dynamically; and (3) any portion of data can be accessed at random.

3 Experimental Results and Discussion

The proposed compression technique is tested on real molecular dynamics datasets. We used two different datasets, as shown in Table 1, for the experiments. These data sets were obtained from snapshots of real molecular simulations. The complete data set had around 600 frames or snapshots. The measurements stored in the data are: x, y and z coordinates of the atoms, charge and mass measured during the simulation.

Table 1. Compression of MD simulation datasets, using PCA and DCT (W = 128)

Data set	Objects	Data Size	Compressed	Ratio	RMSE (Å)
Protein	286,849	3.8 GB	293.45 MB	13.26	0.14
Collagen Fiber	891,272	6.4 GB	535.86 MB	12.23	0.09

The effect of compression on quality of the data is measured as the error in final decompressed data. The error is measured in terms of root mean square error (RMSE; using equation in Table 2). Distance between original data frame F_u and the compressed data frame F_c (we call the data obtained after *decompression* as compressed) is used to compute the error. N_a is the number of atoms in every frame, and N_f is the total number of frames in the data set. We computed the RMSE on locations of objects present in the system.

Table 2. Equations required in experiments and analysis of results

$Error = \sqrt{\frac{1}{N_a \times N_f} \sum_{i=1}^{N_f} \sum_{j=1}^{N_a} \{F_c - F_u\}}$	$MSD(t) = \frac{1}{N} \sum_{i=1}^{N} \lvert \mathbf{r}_i(t) - \mathbf{r}_i(0) \rvert^2$
$MC(t) = \frac{\sum_{i=1}^{N} m_i \mathbf{r}_i(t)}{\sum_{i=1}^{N} m_i}$	$RDF(r) = \frac{N(r)}{4\pi r^2 \delta r \rho}$

The results shown in Table 1 explain the performance of the proposed compression technique. In this method we are able to achieve high compression ratio. The results shown in Table 1 are obtained with a window size of 128. Effect of window size on the compression ratio of the data is shown in Table 3.

Table 3. Effect of window size on Protein data compression

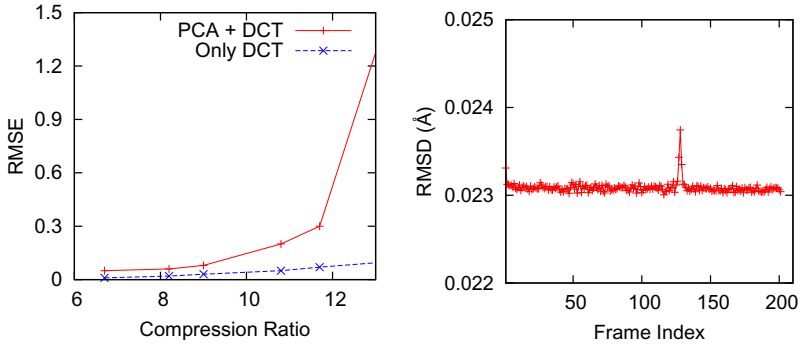
Window Size	Compressed Size	Compression Ratio	RMSE (Å)
008	987.61 MB	3.94	0.25
016	603.29 MB	6.45	0.23
032	432.36 MB	9.00	0.21
064	353.75 MB	11.00	0.18
128	293.45 MB	13.26	0.14
256	386.03 MB	10.08	0.09
512	407.46 MB	9.55	0.05

The compression ratio directly depends on the amount of error that can be allowed to appear in the final data. The errors can be controlled by adjusting coefficients that are retained (as explained in section 2) after the DCT step. The error should increase with the compression ratio, for a fixed window size. This effect is observed in our dataset also. The plot shown in Fig. 3(a) presents this relation between the compression ratio and error. The RMSE increases as the compression ratio is increased. Our PCA and DCT based compression technique gives better compression ratio while maintaining low errors in the compressed data. This method is best suited for compression of very large volumes of data, which are common in the scientific databases.

Data Analysis Results

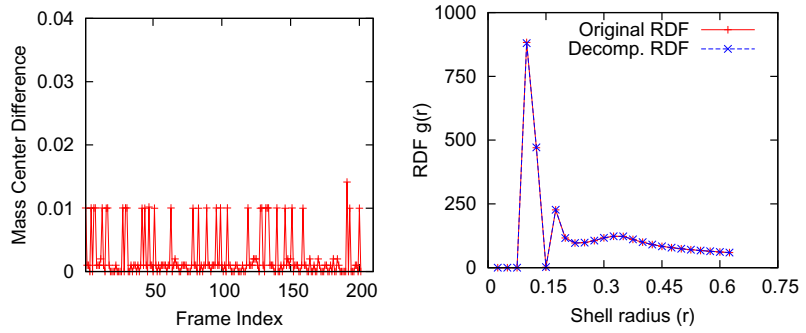
The effect of compression on the data set can also be measured using the results of analytical queries. We applied some queries, interesting to researchers, on the decompressed data to measure this effect.

Mean Square Displacement (MSD): Consider a system with N particles. Let $\mathbf{r}_i(t)$ be the location of particle i, of mass m_i, at time instant t. The MSD information biophysicists ask frequently is formulated by equation given in Table 2. The proposed compression technique introduces very small error in the MSD. The



(a) (b)

Fig. 3. (a) Average RMSE (in Å) plotted against compression ratio. Window size of 128 is chosen 3(b) RMSD (in Å) between original and decompressed frames.



(a) (b)

Fig. 4. Effect of compression: 4(a) On mass centers (in Å). 4(b) Radial distribution function (RDF). Shell resolution $\delta r = 0.025$Å and window size of 128 frames was used.

plot of Fig. 3(b) shows the root mean square displacement of the compressed frame from the original frame. A very small peak in the plot indicates the start of a new window used for the compression. Window size of 128 frames was used for this particular experiment.

Mass Center (MC): Another information of interest to biophysicists is the mass center (MC, Table 2) of the system at any given time instant. Fig. 4(a) shows the effect on mass center, $MC(t)$, of the particle space at a given time instant t. It can be seen that displacement in the mass center is very small. Small fluctuations in the plot are the effects of DCT coefficient quantization. This shows that the proposed method performs well in compression and decompression of the simulation data.

Radial Distribution Function (RDF): Main motivation behind the proposed compression approach is to demonstrate the minimal effect on analytical queries.

We observed this behavior in the experimental results on RDF [9]. The RDF is defined by equation sown in Table 2, where $N(r)$ is the number of atoms in the shell between r and $r + \delta r$ around any particle, ρ is the average density of particles in the whole system, and $4\pi r^2 \delta r$ is the volume of the shell. The RDF can be viewed as a normalized spatial distance histogram (SDH). The SDH is a fundamental tool in the validation and analysis of particle simulation data. Fig. 4(b) shows that RDF is not much affected by compression. For shell resolution $r > 0.025$Å, the RDF values in the compressed and decompressed data are almost same. This shows that the error introduced by compression on RDF of the frames is minimal. Hence, the proposed technique is suited for compression of the molecular simulation data.

4 Conclusions and Future Work

A compression technique for MD data, using PCA and DCT, that can achieve compression ratio of about 13 is presented in this paper. Temporal locality is exploited to achieve good compression that can be used to satisfy the I/O and network bandwidth requirements. Some of the problems that need to be addressed in future are: (1) analysis on compressed data directly; (2) efficient encoding technique, to utilize the correlation between trajectories; and (3) data mining to find patterns of interest in the trajectories. The findings that we reported in this paper will definitely lead to abundant research efforts.

Acknowledgments. This project (R01GM086707) is supported by the National Institute of General Medical Sciences (NIGMS) at the National Institutes of Health (NIH), USA.

References

1. Etten, W.V.: Managing data from next-gen sequencing. Genetic Engineering and Biotechnology News 28(8) (2008)
2. Omeltchenko, A., et al.: Scalable i/o of large-scale molecular dynamics simulations: A data-compression algorithm. Computer Physics Comm. 131(1-2), 78–85 (2000)
3. Ioannidis, Y.E., Poosala, V.: Histogram-based approximation of set-valued query-answers. In: Procs. of VLDB, pp. 174–185 (1999)
4. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets. The VLDB Journal 10(2-3), 199–223 (2001)
5. Salomon, D.: Data Compression: The Complete Reference. Springer (2004)
6. Cochran, W.G.: Sampling Techniques, 3rd edn. John Wiley and Sons (1977)
7. Meyer, T., et al.: Essential dynamics: A tool for efficient trajectory compression and management. Journal of Chemical Theory Computation 2(2), 251–258 (2006)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication (2000)
9. Bamdada, M., et al.: A new expression for radial distribution function and infinite shear modulus of lennard-jones fluids. Chemical Physics 325, 554–562 (2006)